A TRIDENT SCHOLAR PROJECT REPORT

NO. 158

"A Computer Model of Cross-linking Surface Immunoglobulin Receptors on the Surface of a B-Lymphocyte Cell"

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: U.S.N.A. - TSPR; no. 158 (1989)	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): A COMPUTER MODEL OF CROSS-LINKING SURFACE IMMUNOGLOBULIN RECEPTORS ON THE SURFACE OF A B-LYMPHOCYTE CELL.	5. TYPE OF REPORT & PERIOD COVERED: Final 1988/89
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Joseph Francis LePage	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: United States Naval Academy, Annapolis.	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: United States Naval Academy, Annapolis.	12. REPORT DATE: 7 July 1989
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report)
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release; its distribution is UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Accepted by the U.S. Trident Scholar Committee.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Immunoglobulins
B Lymphocytes
B cells
Computer simulation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

While models exist that give information about receptor cluster size for bivalent receptor / bivalent ligand cross-linking, there is currently no detailed data about the cluster shapes. A FORTRAN computer program model which uses a modified Monte Carlo method to simulate the cross-linking of surface immunoglobulin by anti-Ig antibodies was created to provide data on the shapes of the receptor clusters that form. The program logic is summarized in the following statements.

(OVER)

Bivalent receptor sites are randomly placed on a two-dimensional grid. A receptor site is chosen randomly for manipulation. The progabilities of the receptor site becoming unbound, bound to a ligand only, bound to a neighboring receptor, or bound to itself are calculated and calculated and normalized. Through random number selection, the state of the receptor is updated according to the weighted probabilities. The receptor site is then moved by a random angle and distance to simulate the fluid membrane on the surface of a B-cell. The process of choosing sites to be manipulated is repeated and all according updates are made to the master list. Both the duration and the probabilities associated with this process can be controlled as desired. Cross-linked receptors form chains or loops. The shapes of these clusters are expressed mathematically to facilitate their comparison. Analyzing the shape as a function of cluster size and time has shown that the shapes of clusters do not seem to equilibrate. In fact, the variation of shape occurs in the same time frame as the oscillations of intracellular calcium which activate the B-cell. Shape, which goes a step beyond concentration, is a vital link in conforming intramembrane enzymes that control the release of intracellular calcium.

U.S.N.A. - Trident Scholar project report; no. 158 (1989)

"A Computer Model of Cross-linking Surface Immunoglobulin Receptors on the Surface of a B-Lymphocyte Cell"

A Trident Scholar Project Report

by

Midshipman First Class, Joseph F. LePage, Class of 1989

U. S. Naval Academy

Annapolis, Maryland

Boyd A. Waite

Assoc. Prof. Boyd A. Waite - Chemistry

Accepted for Trident Scholar Committee

Dennis F. Hasson

Chairperson

7 July 1989

Date

ABSTRACT

While models exist that give information about receptor cluster size for bivalent receptor / bivalent ligand cross-linking, there is currently no detailed data about the cluster shapes. A FORTRAN computer program model which uses a modified Monte Carlo method to simulate the cross-linking of surface immunoglobulin by anti-Ig antibodies was created to provide data on the shapes of the receptor clusters that form. The program logic is summarized in the following statements. Bivalent receptor sites are randomly placed on a two-dimensional grid. A receptor site is chosen randomly for manipulation. The probabilities of the receptor site becoming unbound, bound to a ligand only, bound to a neighboring receptor, or bound to itself are calculated and normalized. Through random number selection, the state of the receptor is updated according to the weighted probabilities. The receptor site is then moved by a random angle and distance to simulate the fluid membrane on the surface of a B-cell. The process of choosing sites to be manipulated is repeated and all according updates are made to the master list. Both the duration and the probabilities associated with this process can be controlled as desired. Cross-linked

receptors form chains or loops. The shapes of these clusters are expressed mathematically to facilitate their comparison. Analyzing the shape as a function of cluster size and time has shown that the shapes of clusters do not seem to equilibrate. In fact, the variation of shape occurs in the same time frame as the oscillations of intracellular calcium which activate the B-cell. Shape, which goes a step beyond concentration, is a vital link in conforming intramembrane enzymes that control the release of intracellular calcium.

TABLE OF CONTENTS

INTRODUCTION

The immune system consists of lymphocytes, macrophages, a series of macrophage-related cells, and specialized epithelial cells. These cells occur in organized tissues and organs. Lymphocytes and macrophages are also found in substantial quantities in the blood and in the lymph.

Individual lymphocytes are specialized in their commitment to respond to a limited group of structurally related antigens. This commitment exists prior to the first contact of the immune system with a given antigen. The commitment is expressed by binding sites on the lymphocyte membrane which are specific to determinants on that antigen [1].

Lymphocytes differ from one another not only in the specificity of their binding sites, but also in their functional properties. Two broad classes of lymphocytes are recognized: the B-lymphocytes, which are the precursors of antibody secreting cells, and the T-lymphocytes. T-lymphocytes consist of a series of subtypes, some of which mediate important regulatory functions, such as the ability to help or suppress the development of immune responses, including antibody

production [2]. The focus of this study will be T-lymphocyte independent responses only.

The B-cells receive their name from their origin in the bursa of the Fabricius in birds and bone marrow in mammals. The most distinctive and heterogeneous products of B-cells are antibodies, which are also called immunoglobulins (Ig). Several million B-cell clones produce diverse immunoglobulin molecules. Each of the B-cell clones is genetically programmed to make immunoglobulin molecules with unique antigen-binding specificity [3].

Each B-cell produces immunoglobulin molecules with a hydrophobic transmembrane tail and antigen-binding sites facing out from the cell membrane (Figure 1) . These membrane bound antibodies serve as receptors by which antigens can selectively interact with appropriate B-cell clones. Specifically, the antigens cross-link the surface immunoglobulin (sIg) receptors causing conformational changes in the intramembrane enzymes [4] (Figure 1). The modification of these enzymes results in rapid alterations in the physiology of the B-cell which lead to the production of antibodies [5]. Antigenic stimulation can result in B-cell activation, proliferation, and maturation. The proliferation leads to clonal expansion in the form of memory B-cells. Maturation proceeds to a terminal plasma cell stage of differentiation. Plasma

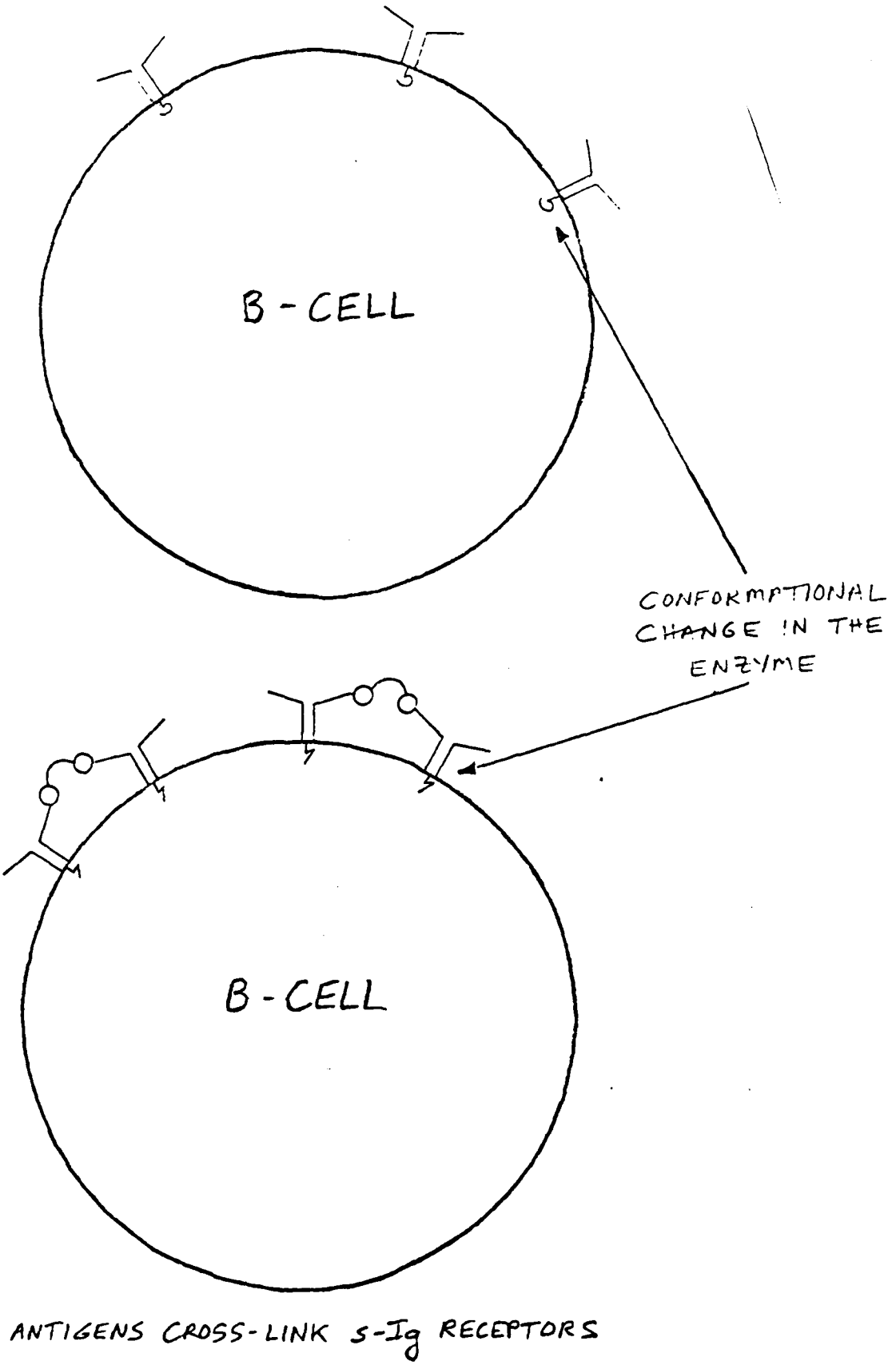

SURFACE IMMUNOGLOBULIN (s-Ig)
B-CELL
CONFORMATIONAL CHANGE IN THE ENZYME
B-CELL
ANTIGENS CROSS-LINK s-Ig RECEPTORS

Figure 1

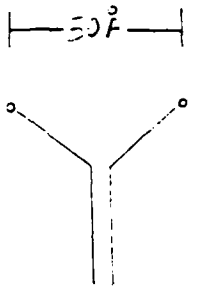
cells are characterized by a high rate of production and secretion of antibodies with hydrophilic tails. The secreted antibodies are otherwise identical with membrane bound antibodies made by the B-cell [3].

ANTIBODY/ANTIGEN VALENCY

Valency is the total number of binding sites present on one antibody or antigen. IgG antibody has two antigen receptor sites (Figure 2) which may be cross-linked. Another type, IgM has a total valency of 10 (Figure 2), but the actual number of binding sites may be less due to steric hindrance [6]. For the purposes of the model to be discussed, bivalent antibodies and bivalent antigens will be used.

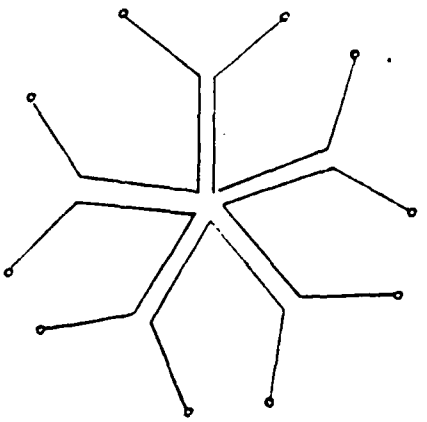
THE ANTI-Ig ANTIBODY

In order to begin a study of B-cell activation, one must choose an appropriate antigen. Anti-Ig antibodies are bivalent antigens with the ability to cross-link surface immunoglobulin (sIg) (Figure 3). In addition to bivalency, anti-Ig antibodies have characteristics which make them suitable. Anti-Ig antibodies, in vivo and in vitro, can rapidly cause changes in B-cell surface binding site expression and



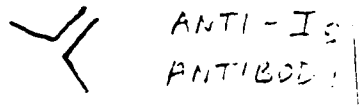
IgG ANTIBODY

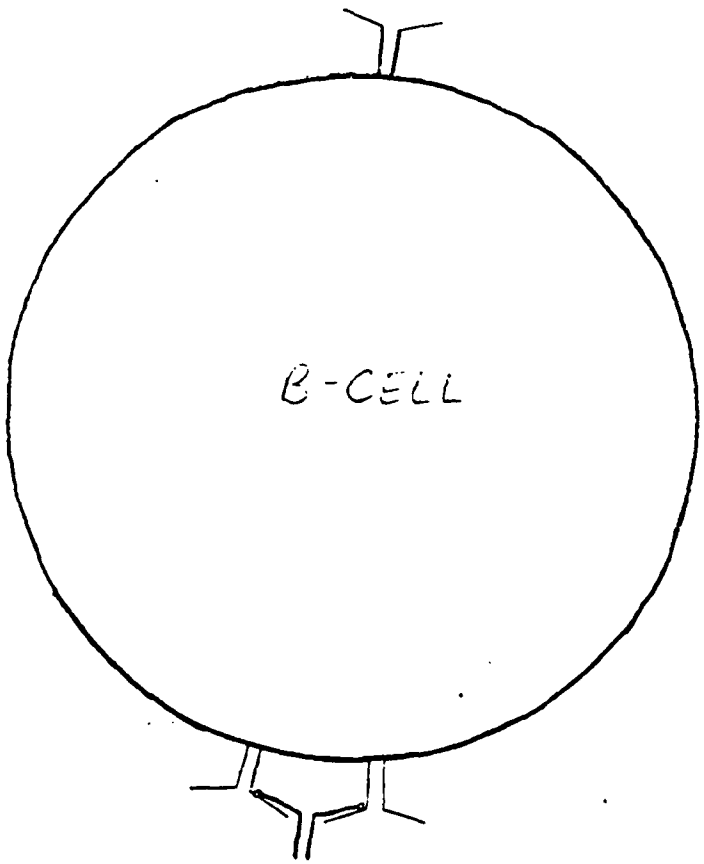
• BINDING SITE



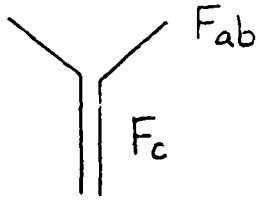
IgM ANTIBODY

Figure 2





ANTI-Ig ANTIBODY CROSS-LINKING s-Ig RECEPTORS



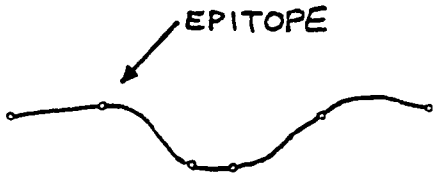
s-Ig BINDING REGIONS

Figure 3

stimulate these cells to synthesize DNA [7]. Since anti-Ig antibodies attach to the more constant binding regions, Fc, of the surface immunoglobulin receptors, the need to purify antigen specific B-cell which have the same variable binding region, Fab, is eliminated [8] (Figure 3). Recall, at this point, that the primary reason for choosing anti-Ig antibodies is their ability to cross-link surface immunoglobulin (sIg).

THE IMMUNON VS. LOW VALENCE ANTIGENS

Two theories currently exist for cross-linking surface immunoglobulin receptors. Dintzis et al. suggest that the B-cell response requires that a minimum of N receptors be connected together by one multivalent antigen, called an immunon (Figure 5) , before an immunological response is generated [9]. An epitope is a binding site on an antigen (Figure 4). They propose that antigens which have at least a threshold number of between 10 and 20 epitopes are stimulatory, while those antigens with fewer are not immunogenic at any dose. Perelson suggests that low valence antigens having even as few as two epitopes can cross-link receptors (Figure 6) and induce activation [9]. Bivalent antigens are known to cross-link receptors to form structures that are linear chains and rings. Such cross-linked structures generate immunological responses



MULTIVALENT ANTIGEN

Figure 4

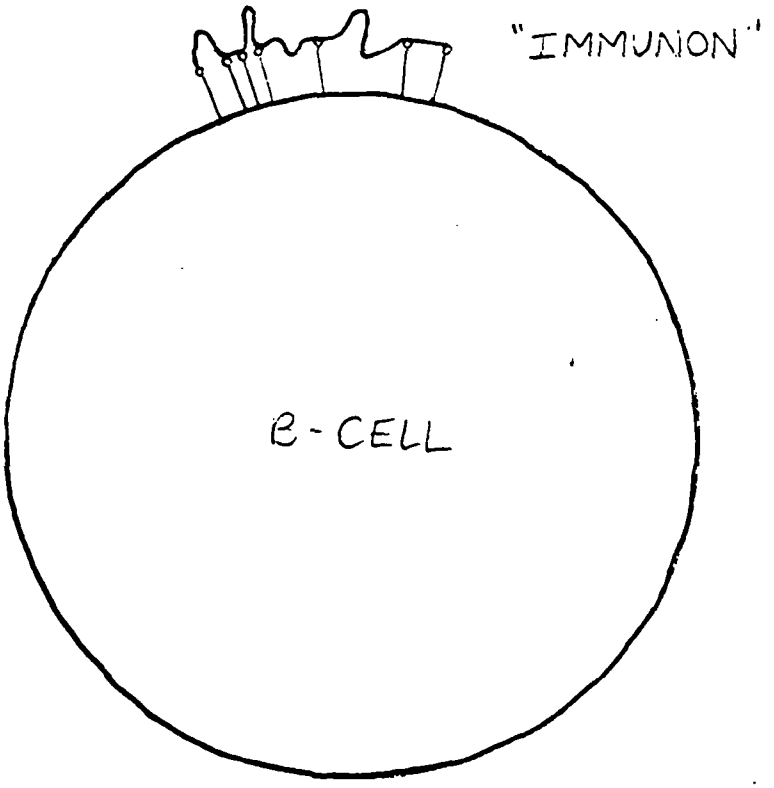
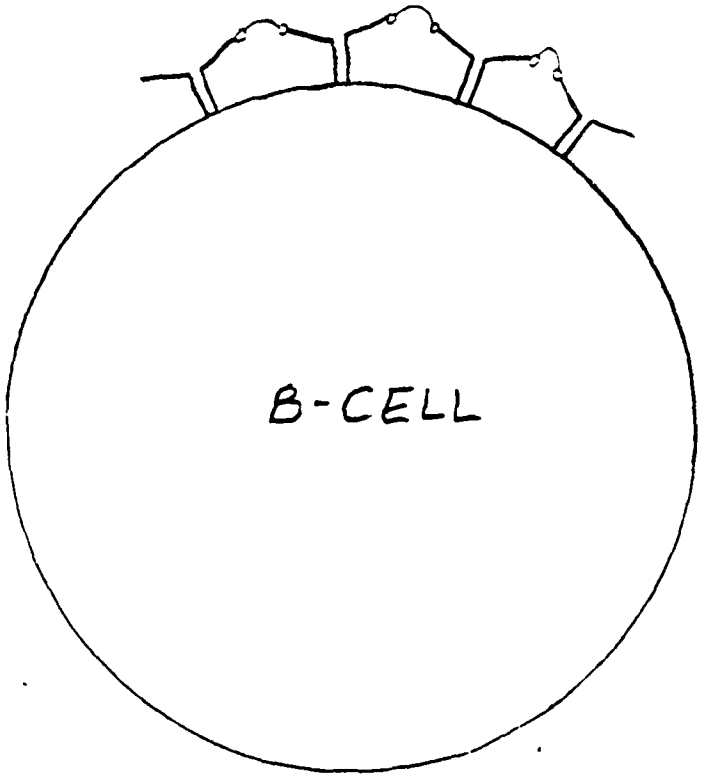


Figure 5



LOW VALENCE ANTIGENS
CROSS-LINKING RECEPTORS

Figure 6

in B-cells, mast cells, and basophils [9].

A multiligand cluster formed from low valence antigens or an immunon formed from a single multivalent antigen accomplish the same effect. That is, both multiligand clusters and immunons cross-link a threshold number of surface immunoglobulin receptors. An important difference arises in the concentration of polymer required for immunogenicity. The amount of bivalent polymer, DNP2-polyethylene oxide, required to stimulate B-cells was 5 x 10-7 M, with peak immunogenicity at 5 x 10-6 M [10]. This concentration is seven orders of magnitude higher than the peak immunogenic concentrations in Dintzis' experiment in which immunons stimulated the B-cells [10]. Possibly, a threshold number of epitopes is needed on antigens at such low concentrations so that significant cross-linking can occur. It seems obvious that low valence antigens would be required in much higher concentrations to link as many receptor sites as an immunon.

While cross-linking is involved in the activation of B-cells, the cluster sizes and the number of clusters necessary for activation are unknown. Based on experiments in which fewer than 10 to 20 epitopes per antigen failed to activate B-cells, the Dintzis theory states that an antigen must have at least N epitopes to be immunogenic [10]. The antigens with the threshold number

of epitopes are called immunons [10].

Although the immunon theory appears to explain the experimental data, Perelson argues that the formation of immunons is not absolutely required. Current theories for cross-linking bivalent receptors with multivalent antigens suggest that clusters of all possible sizes may form. This "theory does not easily support the idea of a critical receptor cluster size being formed only by antigens with greater than a particular number, q, of appropriately spaced haptens" [10]. (Haptens are epitopes that are not bound to an antigen backbone). Additionally, Dintzis' experiments found that the concentration of polymer required for activation was about 7 x 10-13 M for the immunogenic polymers regardless of the number of haptens [10]. The current theory for cross-linking would suggest that hapten concentration and not polymer concentration determines the extent to which receptors are cross-linked [10].

Perelson advocates that a new theory be developed which "incorporates monogamous bivalent attachment of receptors to antigens and concerns itself with making detailed predictions of properties of receptor clusters (e.g., their size, shape, and mobility) as a function of concentration" [11]. There is currently no theory that is capable of addressing details such as shape. The shapes of receptor clusters are significant because they induce

conformational changes in the intramembrane enzymes which lead to B-cell activation.

THE CHEMISTRY INITIATED BY CROSS-LINKING sIg

The foundation has now been set to discuss the chemical pathway. B cell antigen receptor cross-linkage initiates the degradation of phosphatidylinositol-4,5-biphosphate (PIP2), with the resultant formation of two intracellular messengers, diaglycerol and calcium-releasing inositol polyphosphates (IP3) [12] (Figure 7). The diaglycerol operates within the cell membrane to activate protein Kinase-C and the inositol triphosphate causes the endoplasmic reticulum to release Ca++. Diaglycerol, activated protein Kinase-C, and Ca++ are required for B-cell proliferation.

A MONTE CARLO METHOD

Receptor cross-linking results in chains and loops of connected receptor molecules. While ordinary differential equations have been used to model cross-linking, they fell short in several areas. First, the receptors and ligands were assumed to be available to one another without steric restrictions. Because the surface immunoglobulins are attached to the surface of the B-cell, the diffusion rate

The Chemical Pathway of B-Cell Activation

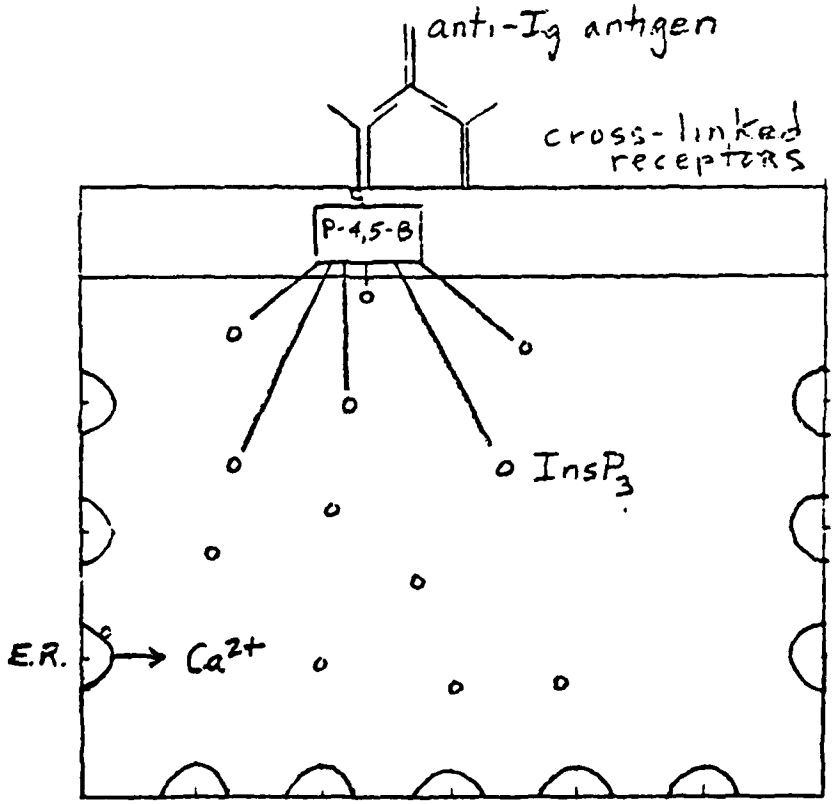


Figure 7

is smaller than assumed in the differential equation approach. Second, the differential equation model gave only the numbers of each type of cluster formed and no information on its shape or spatial dependence. A Monte Carlo approach allows the introduction of the spatial dependence of receptor/ligand interactions and provides detailed information about cluster shape. In a typical Monte Carlo method, a calculation of the number of runs necessary to average out the unrealistic steps would be necessary. An example of this would be a Monte Carlo model of a gaseous system in which intermediate collisions are ignored. However, in the model proposed for cross-linking, the sequence of events for each individual "trajectory" simulate reality [14]. Therefore, calculation of the number of runs needed to average out the implausible steps is unnecessary. In addition, unlike in most Monte Carlo models, an individual run is meaningful. A model based on the statistical averaging of Monte Carlo propagation will provide more information about the nature of cross-linking the receptors on the surface of a B-cell.

AN OVERVIEW OF THE COMPUTER MODEL

Since receptor cross-linking plays a fundamental role in initiating the chemical mechanism of B-cell activation, it will be studied in detail through computer modeling. A quick overview of the assumptions and logic of the program is required.

The following assumptions are made in creating the model. A "patch" of the surface of a sphere can be approximated by a square matrix (Figure 8). Each surface immunoglobulin has two binding sites and antigens are bivalent ligands (Figure 9). It is important to note that only straight chain or closed loop clusters can result from the interaction between bivalent receptors and bivalent ligands (Figure 10).

A general explanation of the program's logic will help to put the more detailed accounts into perspective. The locations of a specified number of binding sites are randomly chosen (Figure 11). One of these receptors is then randomly chosen for manipulation. The neighboring binding sites that are within the radius required for interaction are determined. The state of the binding site to be manipulated and the state of its neighboring binding

A Patch on the Surface of a B-Cell Represented as a Matrix

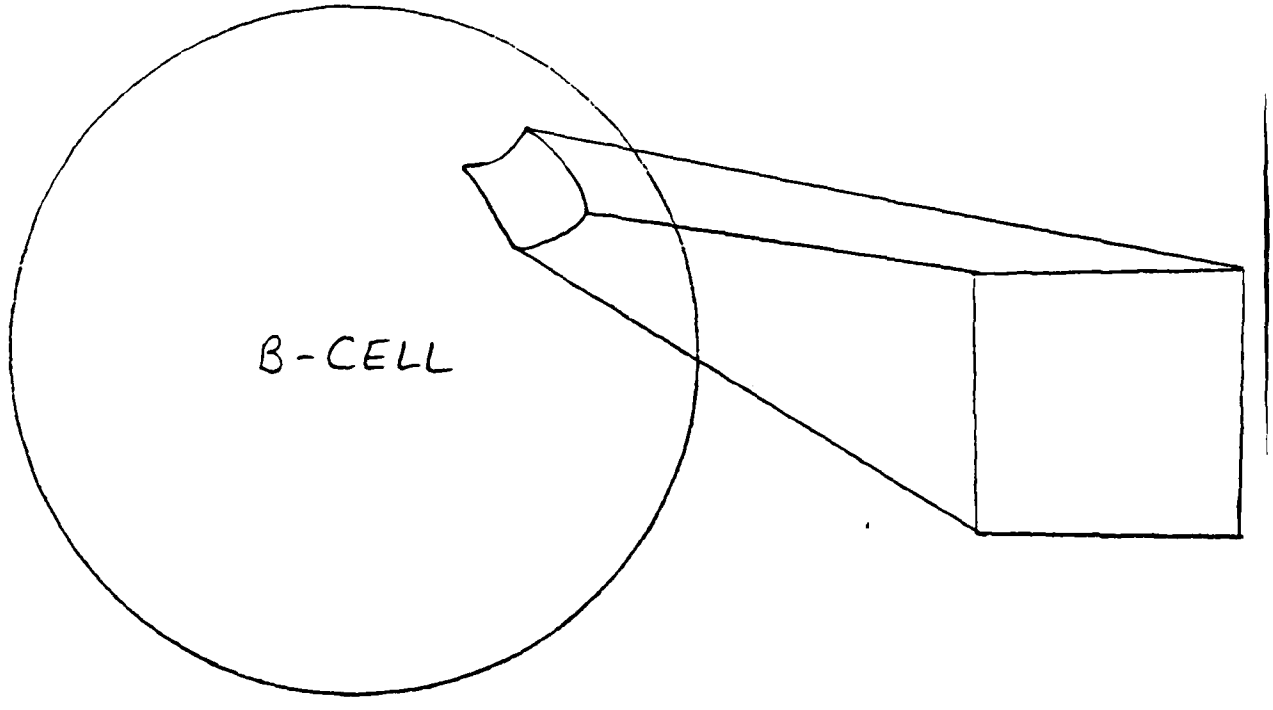


Figure 8

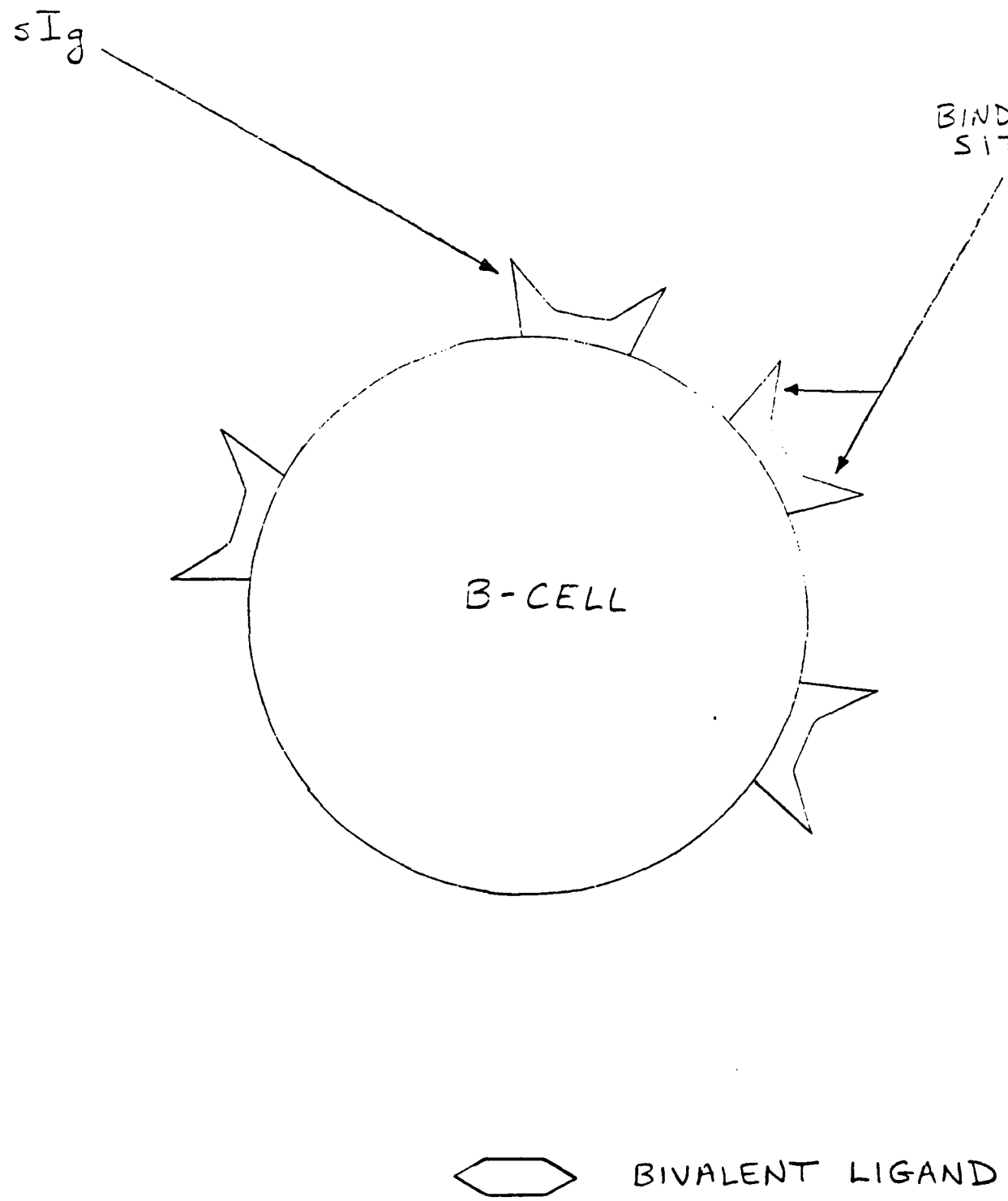
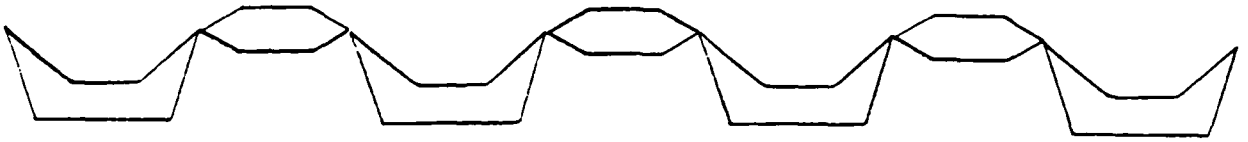
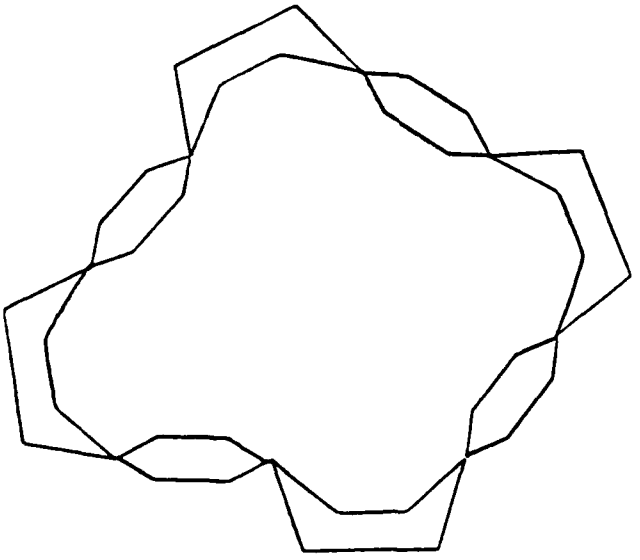


Figure 9

Figure 10



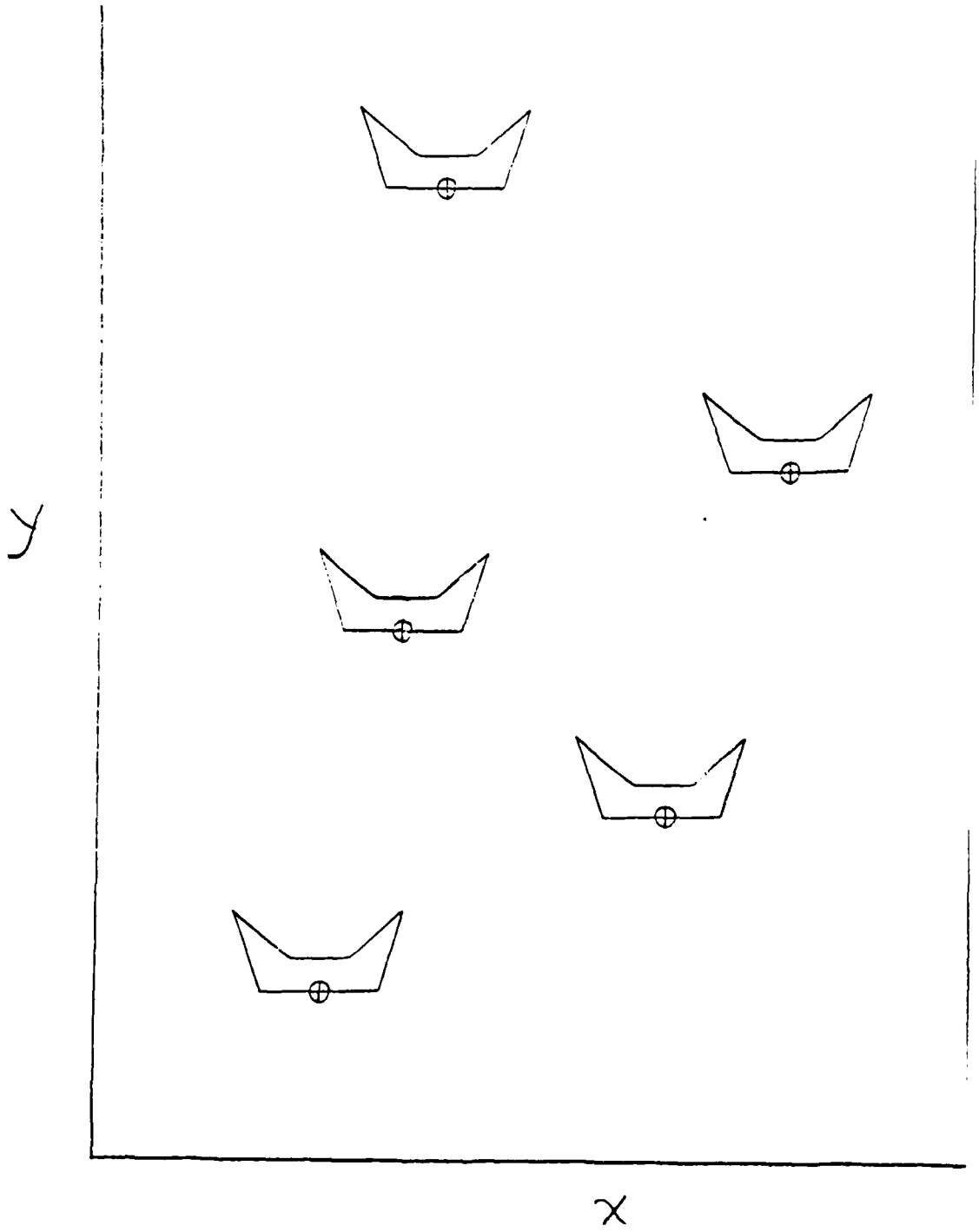
CHAIN



LOOP

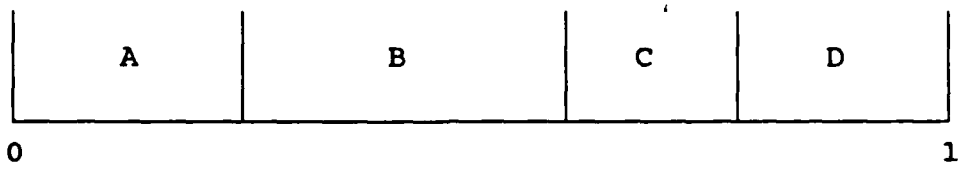
Figure 11

Receptor Site Locations



sites are considered. A binding site may be in one of four states (Figure 12): (0) unbound, (1) bound to a free ligand, (2) bound by a ligand to the other binding site on the same receptor, (3) bound by a ligand to a binding site on another receptor.

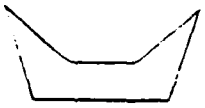
In order to decide how to manipulate the chosen binding site , the program calculates the probability of the binding site achieving each of the four states. These probabilities are based on the number of free ligands, the state of each neighbor, the state of the partner, and the distance to each neighbor. Once calculated, these probabilities are normalized. The algorithm used to decide which manipulation to pursue is pictured below.



Regions A, B, C, and D represent the four states of a binding site. The length of each region is determined by the normalized probability of the chosen binding site becoming that state. A random number between zero and one is selected. The chosen binding site is assigned the state that corresponds to that region. Now, the receptor corresponding to the chosen binding site is moved by some angle and distance that are selected randomly within

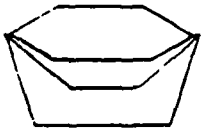
CONDITION

STATE



0

1



2

3

Figure 12

certain boundaries. All of the necessary updates to the master list that accompany this alteration are processed. The program then chooses another of the binding sites at random to manipulate.

The features of the program make it a suitable model for the cross-linking patterns that form on the surface of a B-lymphocyte. Each receptor in the program has two binding sites on it, because the receptors on a B-cell are bivalent (e.g. sIg). The ligands in the model are bivalent (e.g. anti-Ig antibodies) so that they may link two binding sites together. This allows chains of cross-linked receptors to form as has been observed on the surface of B-lymphocytes. The binding sites in the model are initially placed at random in the two-dimensional plane. As a binding site is chosen at random for manipulation, the receptor is also moved randomly within certain angular and radial brackets. The cascade of random selections coupled with movement approximate the fluid membrane in which the receptors on the surface of the B-cells float.

CONNECTION BETWEEN DIFFERENTIAL EQUATION AND MONTE CARLO APPROACH

Two types of reactions must be considered in this discussion. First order and second order events occur during the processes associated with cross-linking. For example, ligand detachment is a first order reaction which can be represented by:

$$[LR] \longrightarrow [L] + [R]$$

$$\text{Rate} = k\,[LR]$$

where [LR] is the probability of selecting an appropriate site and "k" is the probability of the event of detachment occurring. Since detachment will rarely occur if the number of binding sites attached to a ligand only is small, concentration is accounted for implicitly in the selection process.

An example of a second order reaction is a receptor binding to a neighboring receptor. The process can be represented by the following equations:

$$[LR] + [R] \longrightarrow [RLR]$$

$$\text{Rate} = k\,[LR]\,[R]$$

where [LR] is the probability of selecting an appropriate site, [R] is the number of neighboring receptors available for binding, and "k" is the probability of the event occurring. [LR] is implicit in the selection process and [R] is explicitly counted.

First and second order reactions are the only two types that are needed to describe the process of cross-linking in a bivalent ligand / bivalent receptor system.

A DETAILED RUN OF THE COMPUTER MODEL

An example of how the program works in more detail follows. The actual computer program source code is included in Appendix A.

(A) Choosing the Sites of Receptors

After dimensioning the arrays that will be used and setting up initial parameters, the program will choose a specified number of random binding sites on a two-dimensional grid that is 900 angstroms along each axis. The number of binding sites to be chosen is set by the value assigned to "nruns." A calculation of the appropriate number of binding sites to choose follows.

The values chosen for this estimate are within acceptable limits of known values.

$$\text{B-cell radius} = 8 \times 10^{-7}\ \text{m}$$

$$\text{Surface area of B-cell} = 8 \times 10^{-12}\ \text{m}^2$$

$$\text{One B-cell} = 1 \times 10^{5}\ \text{receptors}$$

$$\text{Area / receptor} = 8 \times 10^{-17}\ \text{m}^2$$

$$\text{Area / 100 receptors} = 8 \times 10^{-15}\ \text{m}^2$$

$$\text{Side} = 9 \times 10^{-8}\ \text{m}$$

$$\text{Side} = 900\ \text{Å}$$

Each binding site has a unique set of x-y coordinates that are stored in the arrays "xpt" and "ypt."

(B) Making Each Receptor Bivalent

Since each receptor has two binding sites in this model, a duplicate of each binding site that is chosen at random above is made and assigned the subsequent position in the array. The result is that the first bivalent binding site occupies position 1 and 2 in the "xpt" and "ypt" arrays. The two binding sites on the same receptor will subsequently be referred to as partners. A ligand can thus connect position 1 to another binding site and

another ligand may connect position 2 to still another binding site. The ability of a binding site to be connected to two other binding sites allows receptors to be connected in chains that will vary in length and shape.

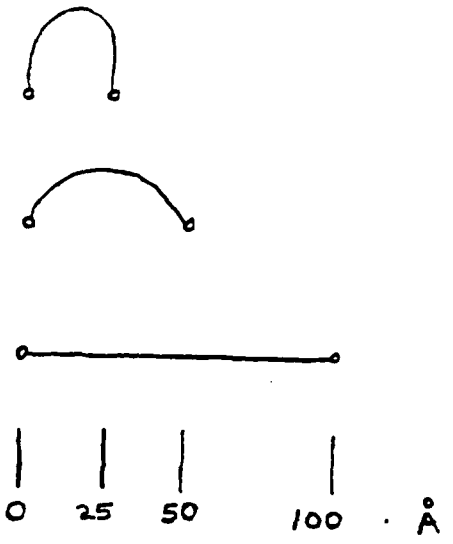
(C) Filing the State of Each Site

An array called "nvalue" is created to hold the state of each binding site. A binding site may be in one of four states (Figure 12): (0) unbound, (1) bound to a free ligand, (2) bound by a ligand to its partner, (3) bound by a ligand to another binding site on a different receptor. It is important to note that for state (3) receptors, a way of recording which receptors are connected to each other is necessary. This is accomplished by assigning pairs of receptors that become cross-linked a state N, where N has an initial value of three and is indexed by one for each successive pair. Since states greater than three exist only as an accounting device, all states greater than three are still treated as state 3 receptors when probabilities are determined. In the rest of this report, the words 'zero', 'one', 'two', and 'three' when used alone will refer to the only possible states of binding sites.

(D) Choosing a Receptor to Manipulate

A position number, L, that corresponds to one of the binding sites is chosen at random. This binding site will be considered as a candidate for manipulation. Two lists of neighboring binding sites will be created. First, the positions of neighboring binding sites that are within 100 angstroms of binding site L are stored in an array called "neigh." The states of these neighbors are held in the array "nval." The distances of each of these neighbors from binding site L are kept in the array "dis." Second, the positions of neighboring binding sites that are within 30 angstroms of binding site L are stored in an array called "nneigh." The states of these neighbors are held in the array "nnval" and the distances of each from binding site L are held in the array "ddis." Receptors that are within 100 angstroms are the only candidates for binding to binding site L because this is the limit of the ligand length (Figure 13). The binding sites within 30 angstroms will be employed in choosing a direction for the movement of this receptor later in the program.

Figure 13



BIVALENT LIGAND

(E) Analyzing the States of the Neighboring Receptors

The states of the neighbors that are within 100 angstroms of binding site L will influence how binding site L may be manipulated. It is useful to know how many of these neighbors are in each of the four states. The number of neighbors in state 0, 1, 2, and 3 are stored in "n0", "n1", "n2", and "n3", respectively.

(F) Determining the Probability of Becoming Each State

A summary of the probability parameters used in the following four sections is given in Figure 14.

(1) The Probability of Becoming a State Zero Receptor

Subroutines are used to determine the probability of binding site L becoming a state 0, 1, 2, or 3 binding site. Function Zero is the probability of binding site L becoming state 0. If binding site L is in state 0, its probability of staying in state 0 is C_a. If binding site L is in state 1, its probability of becoming state 0 is

Probabilities of a Receptor Site Becoming Each of the Four States

Before \ After:	0	1	2	3
0	C_a	C_e*nfree	C_i (if partner nvalue=1)	$\sum$ f(dis) neighbors with nvalue=1
1	C_b	C_f	C_j (if partner nvalue=0)	$\sum$ f(dis) neighbors with nvalue=0
2	C_c	C_g	C_k	0
3	C_d - f(dis)	C_d - f(dis)	0	C_h

Figure 14

C_b. This is a first order rate process as discussed earlier. If binding site L is in state 2, its probability of becoming state 0 is C_c. If binding site L is in state 3, its probability of becoming a zero is given by expression (1) below and is graphed in Figure 15:

$$C_d\text{-}0.8*\exp(-0.0003*(\text{dis(ne)}-50)^2) \qquad (1)$$

where "ne" is the position of the binding site to which binding site L is currently linked. This function was chosen because the probability of the ligand detaching from binding site L should increase as the distance between the receptors deviates from the optimal distance due to ligand strain.

(2) The Probability of Becoming a State One Receptor

Function One is the probability of binding site L becoming a state 1 binding site. If binding site L is in state 0, its probability of becoming a state 1 binding site is given by expression (2):

$$C_e*\text{nfree} \qquad (2)$$

where "nfree" is the number of free ligands still

available for binding to binding sites. This is an example of the second order rate law discussed earlier. If binding site L is in state 1, its probability of staying a state 1 binding site is C_f. If binding site L is in state 2, its probability of becoming a state 1 binding site is C_g. If binding site L is in state 3, its probability of becoming a state 1 binding site is given by expression (3):

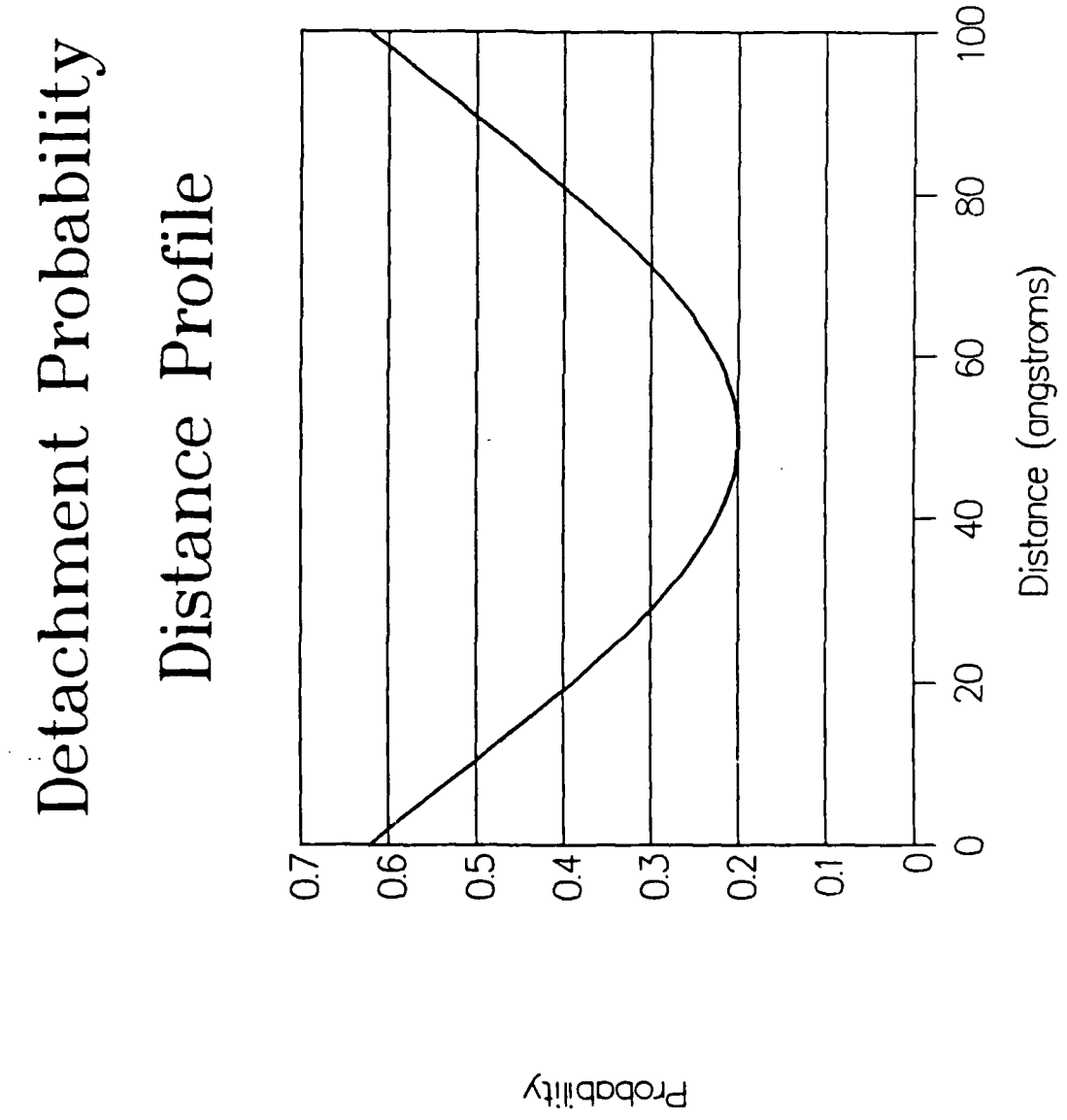
$$C_d\text{-}0.8*\exp(-0.0003*(\text{dis(ne)}-50)^2) \qquad (3)$$

where "ne" is the position of the binding site to which binding site L is bound.

(3) The Probability of Becoming a State Two Receptor

Function Twp is the probability of binding site L becoming a state 2 binding site. If binding site L is in state 0, its probability of becoming a state 2 binding site is C_i if its partner is in state 1. If binding site L is in state 1, its probability of becoming a state 2 binding site is C_j if its partner is in state 0. If binding site L is already in state 2, its probability of remaining unchanged is C_k. Finally, if binding site L is a state 3 receptor, it has no chance of becoming a state 2

Figure 15


Detachment Probability
Distance Profile
Probability
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
20
40
60
80
100
Distance (angstroms)

receptor directly in one step (See Figure 12).

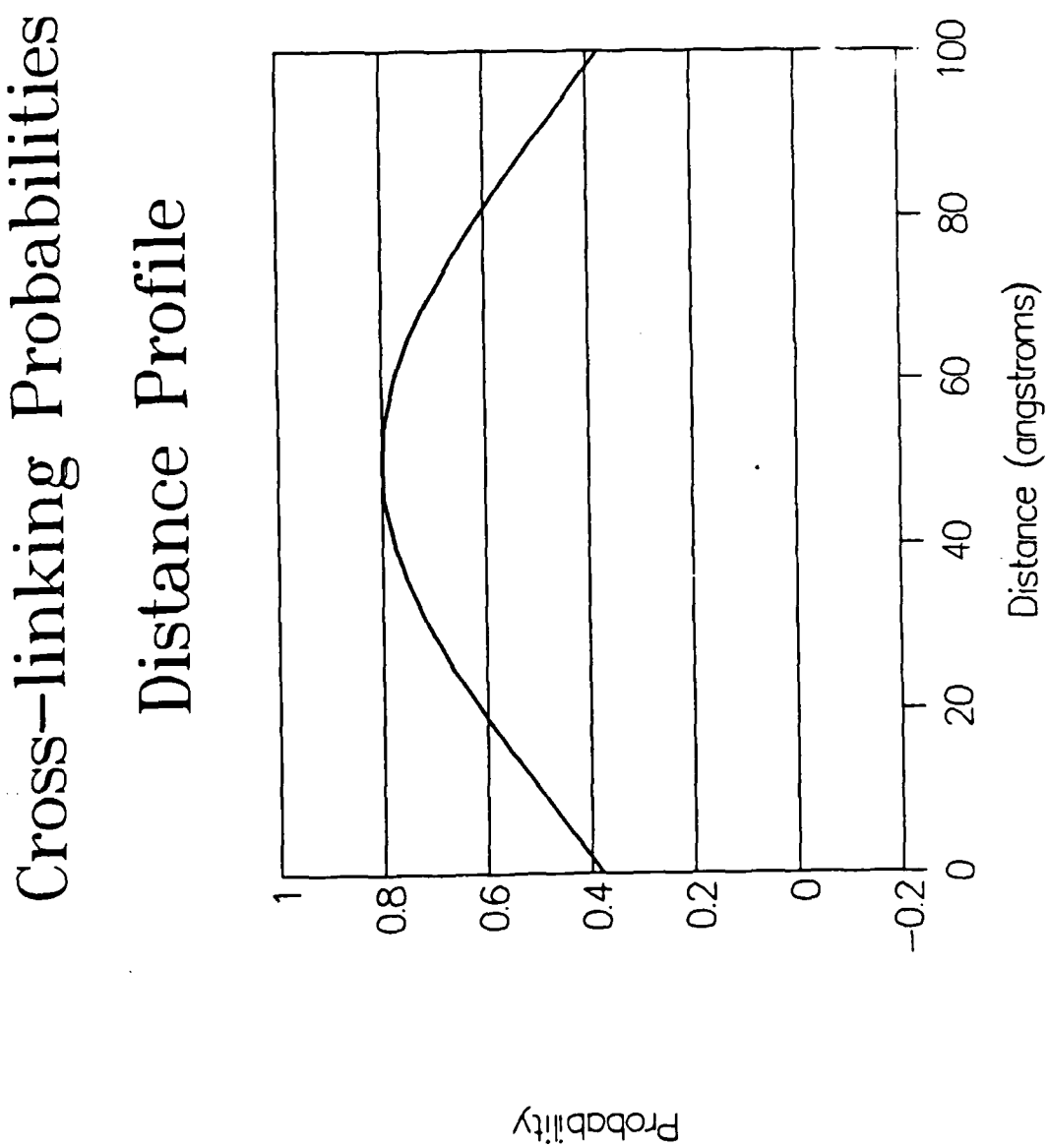
(4) The Probability of Becoming a State Three Receptor

Function Twn is the probability of binding site L becoming a state 3 binding site. If binding site L is in state 2, it has no chance of becoming a state 3 binding site. If binding site L is in state 3, its probability of staying attached to the binding site it is currently attached to is C_h. For binding site L currently in state 0 or 1, another subroutine is employed to sum up the individual probabilities of binding site L attaching to the neighbors within 100 angstroms. Each case will be treated separately. First, if binding site L is currently in state 0, it may only attach to neighboring binding sites that are in state 1. The contribution for each neighbor in state 1 to the probability is given by expression (4) below and is graphed in Figure 15.

$$0.8*\exp(-0.0003*(dis(c)-50)^2) \qquad (4)$$

where "c" is the position of that state 1 neighbor. This function was chosen because the probability of cross-linking receptors should decrease as the distance becomes closer or further than the optimal ligand length.

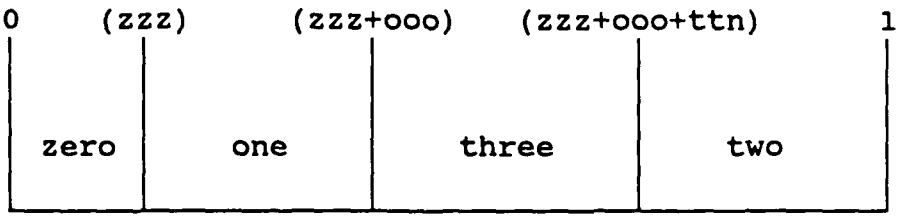
Figure 16


Cross-linking Probabilities
Distance Profile
Probability
1
0.8
0.6
0.4
0.2
0
-0.2
0
20
40
60
80
100
Distance (angstroms)

The sum of this expression for all the state 1 neighbors within 100 angstroms of binding site L gives the probability of that binding site L will go from a state 0 to a state 3 binding site. Second, if binding site L is currently in state 1, it may only attach to neighboring binding sites that are in state 0. The expression will be the same as for the state 1 neighbors, but now only the expression will only be evaluated for the binding sites that are in state 0. The sum of the expression for the state 0 binding sites will give the probability that binding site L will go from state 1 to state 3 binding site.

(G) Normalize the Probabilities

Once the subroutines have determined the probabilities of binding site L becoming a state 0, 1, 2, or 3 binding site, these four probabilities are normalized. The normalized probabilities are "zzz", "ooo", "ttp", and "ttn", respectively. The range of numbers between zero and one will be partitioned into four zones based on these normalized probabilites.



In this chart, the first row gives the values for the partitions and the second row gives the state that binding site L will become if that range is selected. A range is selected by choosing a random number between 0 and 1. The probability of binding site L becoming a two, for example, is weighted by the size of the two range relative to the other ranges. Four distinct sections of the program are devoted to changing binding site L into each of the possible states to which it may be required to change.

(H) Updates Required After the Manipulation

A summary of the updates required for a receptor to go from state x to state y is given in Figure 17.

(1) To Become a State Zero Receptor

This section will be invoked if binding site L is to become a state 0 binding site. If binding site L was a zero, it will remain a zero. If binding site L was a one,

Updates Required to Accomplish a Receptor Site Change of State

To: \ From:	0	1	2	3
0	no change	nvalue(ℓ)=0 nfree=nfree +1	nvalue(ℓ)=0 nvalue (npart)=1	nvalue(ℓ)=0 nvalue(neigh(c)) =1
1	nvalue(ℓ)=1 nfree=nfree -1	no change	nvalue(ℓ)=1 nvalue (npart)=0	nvalue(ℓ)=1 nvalue(neigh(c)) =0
2	nvalue(ℓ)=2 nvalue (npart)=2	nvalue(ℓ)=2 nvalue (npart)=2	no change	not allowed
3	nvalue(ℓ)=3 nvalue (neigh(c)) =3 (DODA Subroutine)	nvalue(ℓ)=3 nvalue (neigh(c)) =3 (DODA Subroutine)	not allowed	no change

Figure 17

it will become a zero and the number of free ligands will be increased by one. If binding site L was a two, it will become a zero and its partner will become a one. If binding site L was a three, it will become a zero and the neighbor to which it was attached will become a one.

(2) To Become a State One Receptor

This section will be invoked if binding site L is designated to become a one. If binding site L was a zero, it will become a one and the number of free ligands will be reduced by one. If binding site L was a one, it will remain a one. If binding site L was a two, it will become a one and its partner will become a zero. If binding site L was a three, it will become a one and the neighbor to which it was attached will become a zero.

(3) To Become a State Two Receptor

This section will be used if binding site L is to become a two. A state 2 binding site is one which is bound to the other binding site on the same receptor by a ligand (Figure 12). If binding site L is a zero or a one, both binding site L and its partner will become twos. If

binding site L is a two, it will remain a two. If binding site L is a three it cannot become a two.

(4) To Become a State Three Receptor

This section will be invoked if binding site L is to become a three. A three is a state in which a binding site is bound to a binding site on neighboring receptor by a ligand (Figure 12). If binding site L was a three, it will remain a three. If binding site L was a two, it cannot become a three. If binding site L was a one, the "doda" array will assign a probability of binding for each of the neighbors (equation 1) that are both within 100 angstroms and state 0 based on distance:

$$doda = 0.8 * \exp(-.0003*(dis(c)-50)^2) \qquad (1)$$

where "c" is the location of the neighbor being considered in the "neigh" array. A graph of equation (1) is shown in Figure 16. The probabilities will be normalized and used as partitions. A random number from 0 to 1 is selected and the the range which corresponds to its value will be the neighboring binding site that is chosen for the binding. Receptor L and the chosen neighbor will become threes. If binding site L is a zero the same procedure as

if binding site L was a one will be followed, except only neighbors whose values are zero will be considered for binding.

(I) Random Walk in Two-Dimensions with Steric Hindrance

After binding site L has been manipulated and the affected binding sites have been updated, binding site L is moved. This movement models the fact that the binding sites on B-lymphocytes are not fixed. A radius and angle for the movement will be selected in the following manner. A random number, "rand" in the range 0 to 1 is chosen. The radius of the move is given by equation (2) below and graphed in Figure 18:

$$\text{radius} = 42 * \text{rand}^3 - 63 * \text{rand}^2 + 31 * \text{rand} \qquad (2)$$

This function was chosen by a standard mapping procedure in Monte Carlo modeling [14]. The desire was to choose a radius for movement in a realistic way. The radius should have the greatest probability of being the average value and the probability of choosing a high or low radius should diminish as the deviation from the average increases. The function pictured conforms to these criteria. The angle of the move will be restricted by the location of neighbors that are within 30 angstroms. The

Step-size Selection for Random Walk

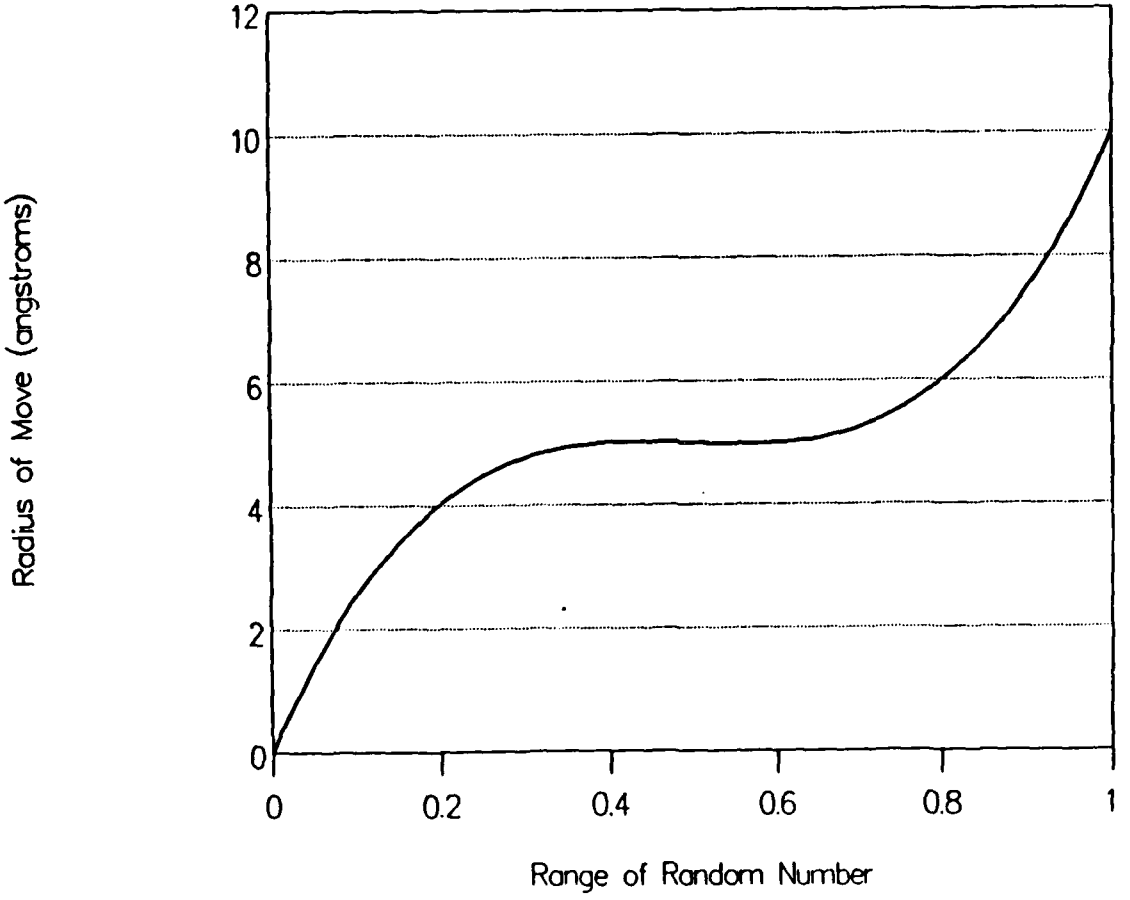


Figure 18

angle of each of these neighbors plus or minus 0.10 radians will be excluded from the possible angle of the move. The possible angle of the move is then chosen at random. If binding site L is a three and moving it by the angle and radius determined above will result in binding site L being greater than 100 angstroms from the neighbor to which it is attached, the movement will not be allowed.

SUMMARY OF ONE ITERATION

The manipulation of binding site L and the appropriate updates are now complete. The next iteration of the program will pick another binding site L at random and perform the same process. The program will continue until "nlpoh" iterations have been completed. Upon performing "nlpoh" iterations the dynamic portion of the program is finished. It is important to relate one iteration of the program to a unit of real time. The following derivation is provided toward that purpose. From gas kinetic theory (which is justifiable in that it approximates the liquid state in terms of the effective number of "encounters" as derived from liquid kinetic theory) [15], the number of collisions per second in a unit area, Z_w, is expressed by:

$$Z_w = (1/4)(c)(N/V) \quad (1)$$

where c is the average speed of a particle and N is the number of particles in volume V. An expression for the mean free path is given by:

$$\lambda = (1/\sigma)(V/N) \quad (2)$$

where σ is the effective cross-sectional area of a receptor and N is the number of particles in volume V. The diffusion constant, D, is determined by using the following equation:

$$D = (1/3)(\lambda)(c) \quad (3)$$

Combining equation (2) and equation (3) gives the following expression for the diffusion constant:

$$D = (1/3)(1/\sigma)(V/N)(c) \quad (4)$$

Substituting equation (4) into equation (1) yields an expression for the number of collisions per second in a unit area, Zw:

$$Zw = (3/4)(D)(\sigma)(N/V)^2 \quad (5)$$

The commonly accepted value for D for macromolecules the size of antibodies is 10^{-7} cm^2/sec. The cross-sectional area, σ, is estimated to be $(100 \times 10^{-6} \text{ cm})^2$. The value of N/V for a 10^{-6} M concentration is $(10^{18}$particles $/10^3$ $cm^3)$. Substituting these values into equation (5) gives:

$$Z_W = (3/4)(10^{-7} \text{ cm}^2/\text{sec})(100 \times 10^{-6} \text{ cm})^2 \times (10^{18}/10^3)^2 \text{cm}^{-6}$$

Evaluating the above equation yields

$$Z_W = 2.4 \times 10^{11} \text{ collisions/sec-cm}^2 \qquad (6)$$

Equation (6) must be multiplied by a conversion factor to find the number of collisions per second in a square angstrom.

$$Z_W = (2.4 \times 10^{11} \text{ collisions/sec-cm}^2)(10^{-16} \text{ cm}^2/\text{Å}^2) \qquad (7)$$

Equation (7) is multiplied by the patch area in square angstroms to give the number of collisions per second:

$$\text{collisions/sec} = (2.4 \times 10^{-5} \text{ hits/sec-Å}^2)(7.84 \times 10^6 \text{ Å}^2)$$

Evaluating the above expression yields:

collisions/sec = 190 [on the total patch]

To calculate the number of hits per second that are actually collisions between ligands and receptors, as opposed to ligand collisions with bare cell membrane, the total patch must be multiplied by a ratio of the effective receptor area vs. the total patch area.

good hits/sec = (190)(area of receptors/area of patch)

good hits/sec = $(190)(3.14 \times 10^4 / 7.84 \times 10^6)$

The result is the number of "good" or potentially interactive hits per second is:

good hits/sec = 0.761

Since one good hit corresponds to one iteration of the program, it can be concluded that:

One iteration corresponds to 1.31 seconds of real time.

MAKING A PICTURE OF THE CHAINS

The graphical section employs the following sorting logic. Each receptor that is the beginning or the end of a chain will have a binding site that is a zero or a one. The array "nvalue" is gone through systematically. Each time a binding site that has a zero or a one is found it is considered to be the beginning of a chain. If its partner's value is a three or greater, the array "nvalue" is searched for a binding site with a matching value (the value which identifies it to be the binding of the other end of the same ligand). This binding site's partner's value is then determined. If it is a zero or a one that is the end of the chain. If the value is a three or greater, the next link in the chain is sought. The location of each of the receptors that make up this chain are output to a two-dimensional array called "grph." The following example should be referred to while reading the subsequent discussion:

"GRPH" ARRAY

Row	Column			
	1	2	3	4
1	50	39	23	22
2	26	12	19	6
3	0	0	27	43
4	0	0	0	0

For the first chain, the x and y locations of the first receptor will be in column 1 and 2, respectively, of row 1. The x and y locations of the second receptor will also be in column 1 and 2, respectively, but in row 2. More rows will be occupied until the first chain is completed. The x and y locations of the receptor of the second chain will be in columns 3 and 4, respectively for as many rows as it takes to complete the chain. It should be noted that as a chain is being created and sent to the appropriate columns and rows of the "grph" array, the "nvalue" for those binding sites are changed to zero since they have been accounted for. When the "nvalue" array has

been systematically completed, all the chains have been entered into the "grph" array. The "nvalue" of all the binding sites that make up the chains have all been set to zero.

MAKING A PICTURE OF THE LOOPS

The only non-zero "nvalues" remaining are the twos and the threes (or greater) that are part of closed loops. The twos cannot be part of closed loops so they will be ignored. This current "nvalue" list is approached systematically. A three or greater is found and considered the beginning of a chain. Receptors that are linked successively are found by tracing matching binding site nvalue's. When the receptor that was considered the beginning is found as a successive link the closed loop is complete. The receptor locations have been sent to the "grph" array the same way the receptor locations for the chains were. Likewise, as a chain was being formed, the nvalues for those receptors were set to zero so that the same closed loop will not be recorded more than once from different starting points. When complete, the closed loops will have all been recorded once and the "nvalue" array will only contain zeros and twos. A one-dimensional array "nchl(n)" stores the number of chains having length n. For example, if four chains each of which links three

receptor exists, nchl(3) = 4. The number of chains of each type of length are tallied as the chains are found and sent to the "grph" array. When multiple runs of the program are done for statistical verification, the one-dimensional "nchl" arrays become the columns in the two-dimensional array called "nstatu." This arrangement fascilitates the comparison of data between runs.

SHAPES OF RECEPTOR CLUSTERS

The program has provided the size and shape details of the cross-linking patterns which result from bivalent ligand / bivalent receptor interactions. With the discovery of such data, new concepts in the elucidation of the B-cell activation which go beyond simple concentration of the antigen are possible. The intricacies of the shapes of clusters may be a vital link in the B-cell activation process.

MATHEMATICALLY EXPRESSING SHAPE

Characterizing the shapes of the chains mathematically allows the chains to be compared as the parameters of the program are altered. A technique that eloquently accomplishes the task of mathematically expressing shapes is found in the work of Rudnick and Gaspari on the shapes

of random walks [13].

The first quantities to be calculated are the eigenvalues of a "d x d" tensor, T, called the radius of gyration tensor and defined by:

$$T_{ij} = 1/N \sum_{1}^{N} (x_{1i} - \langle x_i \rangle)\ (x_{1j} - \langle x_j \rangle)$$

where the object in question is assumed to consist of N parts, the 1th of which is located at x_1. The radius of gyration matrix for our two-dimensional surface is expressed by:

$$T_{ij} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} .$$

The eigenvalues of the radius of gyration matrix are given by the following equations:

$$R1 = ([(A + D) + \{(A + D)^2 - 4(AD - CB)\}^{0.5}] / 2)^{0.5}$$
$$R2 = ([(A + D) - \{(A + D)^2 - 4(AD - CB)\}^{0.5}] / 2)^{0.5}$$

A measure of shape is provided by the quantity, A_d, called the asphericity or the asymmetry which is mathematically expressed by:

$$A_d = ([R1^2 - R2^2]^2) / ([R1^2 + R2^2]^2)$$

for a two-dimensional system. The asphericity, Ad, has zero as its lower bound when the pattern is spherical, and one as its upper bound when the object is one-dimensional. Asphericity is an excellent one-parameter measure of the shape's average deviation from sphericity.

ASPHERICITY VS. CHAIN LENGTH

The asphericity ranges in value from zero to one. An asphericity value of one indicates the cluster is only one-dimensional. If the asphericity value is less than one, the cluster becomes less and less elongated as zero is approached. At zero, the cluster is as long as it is wide in two-dimensions.

In this analysis of cross-linking, the average asphericity will be plotted as a function of chain length. The average asphericity for each chain length is determined by summing the asphericities for chains of that length and dividing by the number of chains that were summed. Two runs of the program in which only the cross-linking probability function is changed were

performed. In the first run the function is:

$$0.8 * \exp(-0.0009*(dis(c)-50)^2) \qquad (1)$$

In the second run the cross-linking probability function is:

$$0.8 * \exp(-0.0003*(dis(c)-50)^2) \qquad (2)$$

It is apparent that only the constant term in the part of the function that has been raised to a power has been altered. A picture of these two functions is provided in Figure 19. The highest probability of cross-linking occurs when the receptors are at a distance of 50 Angstroms. Function (2) makes the probability of receptors cross-linking higher with neighbors that are closer and further than the optimal distance.

For each of the two functions, an 800 iteration run was performed (Figures 20, 21). During each run, a graph of the asphericity versus the chain length was made after every 200 iterations (Figures 22-29) . These graphs can be compared to elucidate the cross-linking process.

The graphs that result after 200 iterations (Figures 22, 26) will be considered first. For both functions, the elongation of the chain generally decreases as the length of the chain increases. A significant difference occurs in the average illongation of the chains having three

Cross-linking Probability Distance Profile

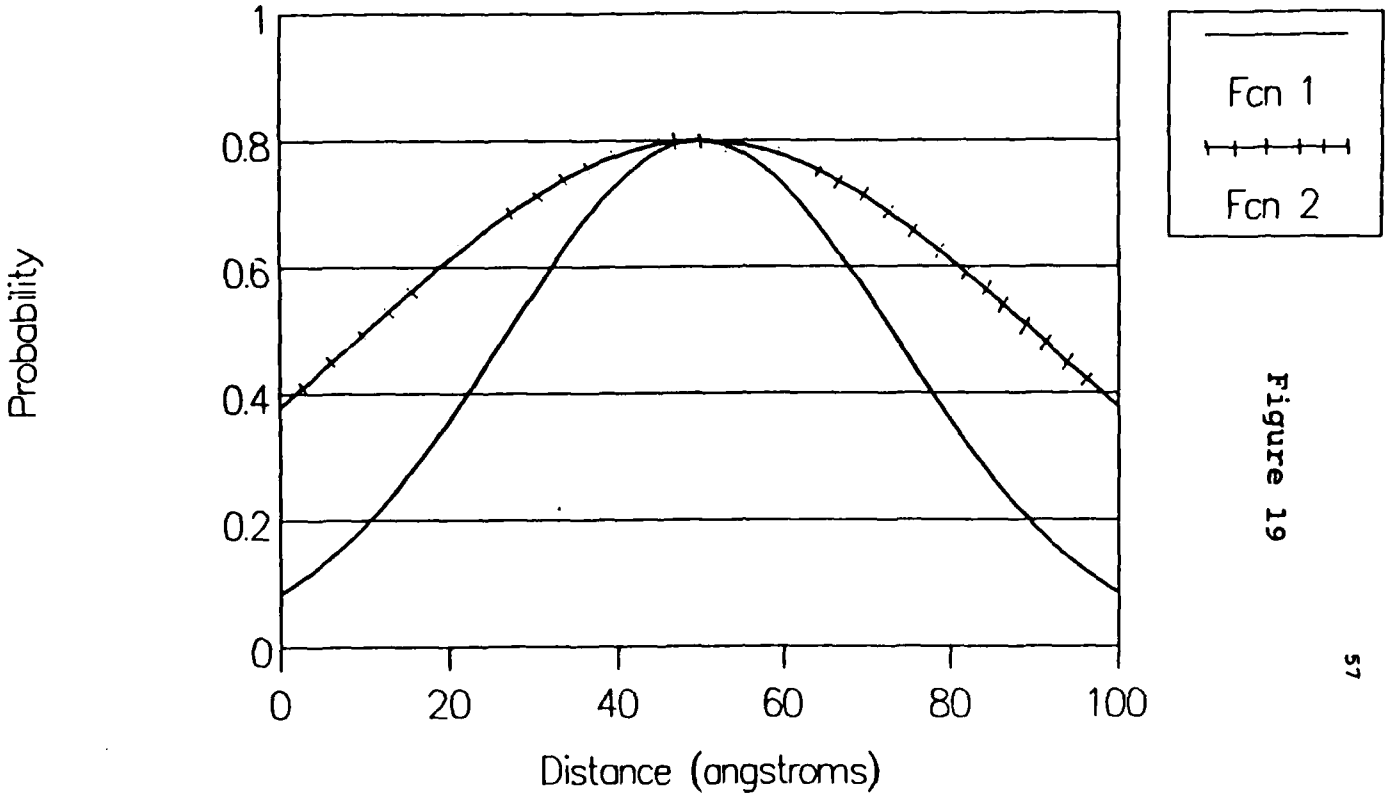


Figure 19

TABLE: FUNCTION (1)

Chain Length	Asphericity			
	Number of Iterations Completed			
	200	400	600	800
2	1.00	1.00	1.00	1.00
3	.849	.933	.655	.672
4	.728	.665	.102	.513
5	-	.890	.567	.303
6	.076	.417	.687	-
7	-	.835	-	-
8	-	-	-	-

Figure 20

TABLE: FUNCTION (2)

Chain Length	Asphericity			
	Number of Iterations Completed			
	200	400	600	800
2	1.00	1.00	1.00	1.00
3	.457	.730	.751	.983
4	.767	.702	.628	.674
5	.690	.692	.537	-
6	-	-	.404	-
7	-	.748	.701	.596
8	-	.658	-	-

Figure 21

Figure 22

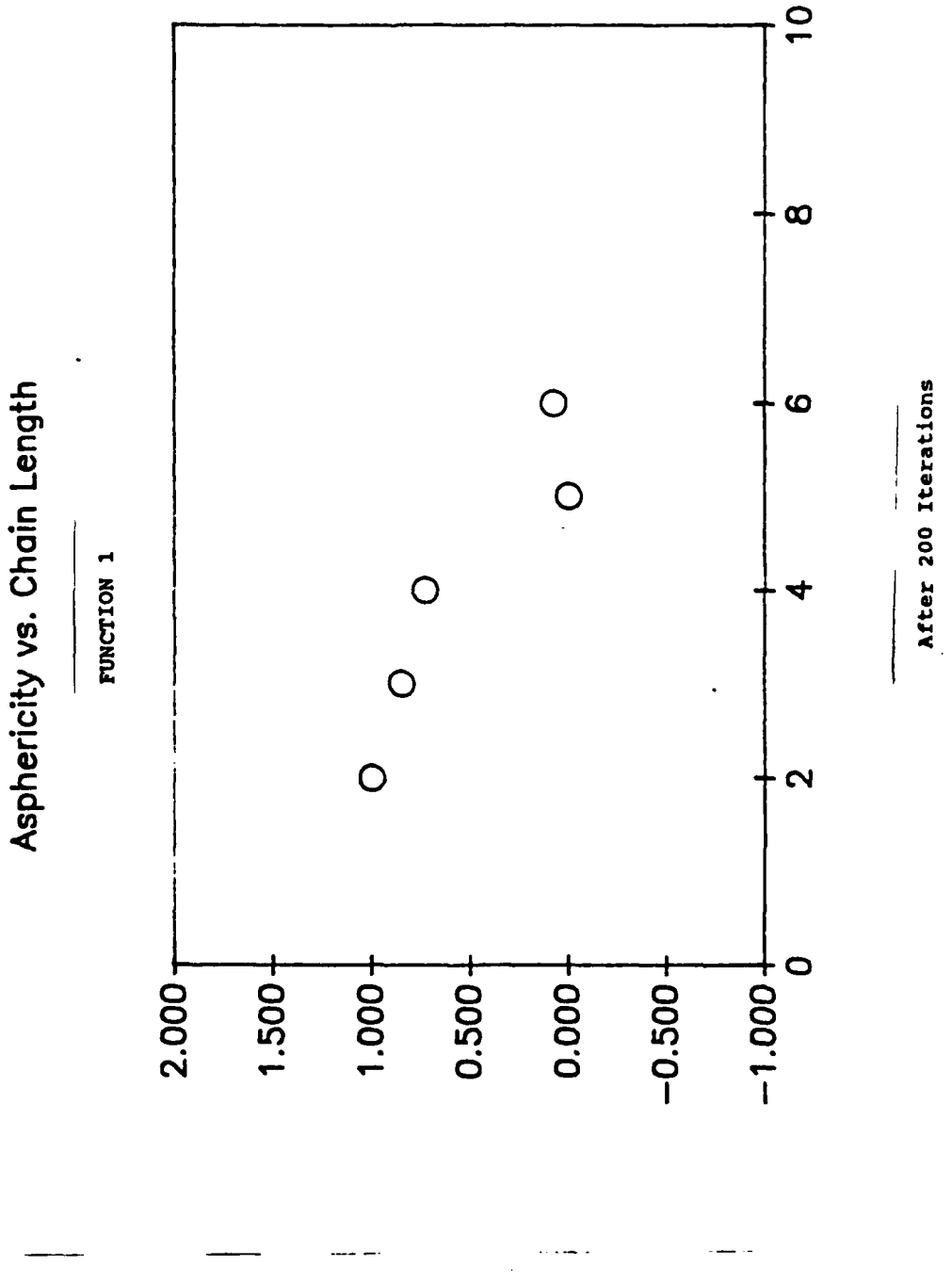

Asphericity vs. Chain Length
FUNCTION 1
After 200 Iterations
2.000
1.500
1.000
0.500
0.000
-0.500
-1.000
0
2
4
6
8
10

Figure 23

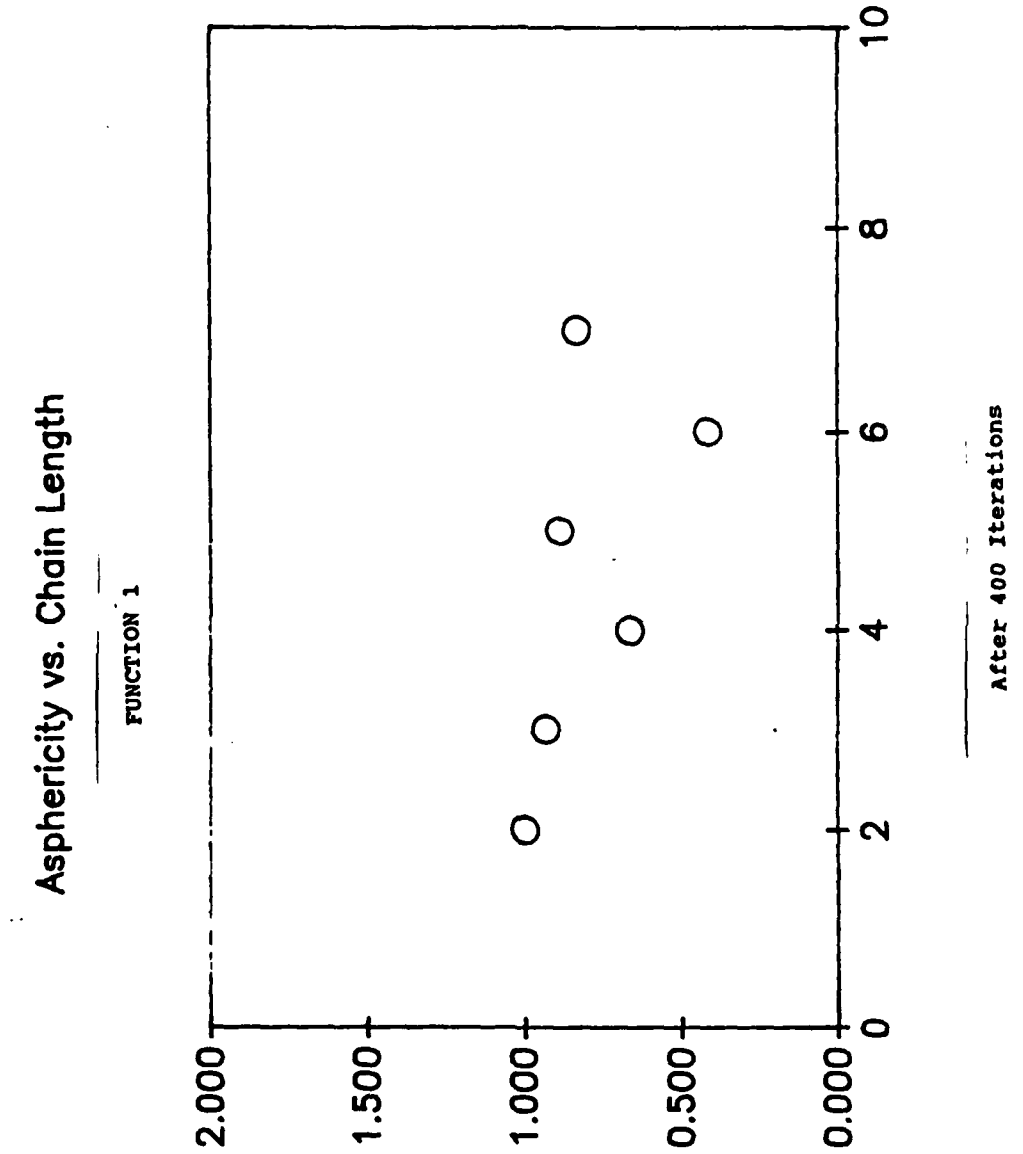

Asphericity vs. Chain Length
FUNCTION 1
2.000
1.500
1.000
0.500
0.000
0
2
4
6
8
10
After 400 Iterations

Figure 24

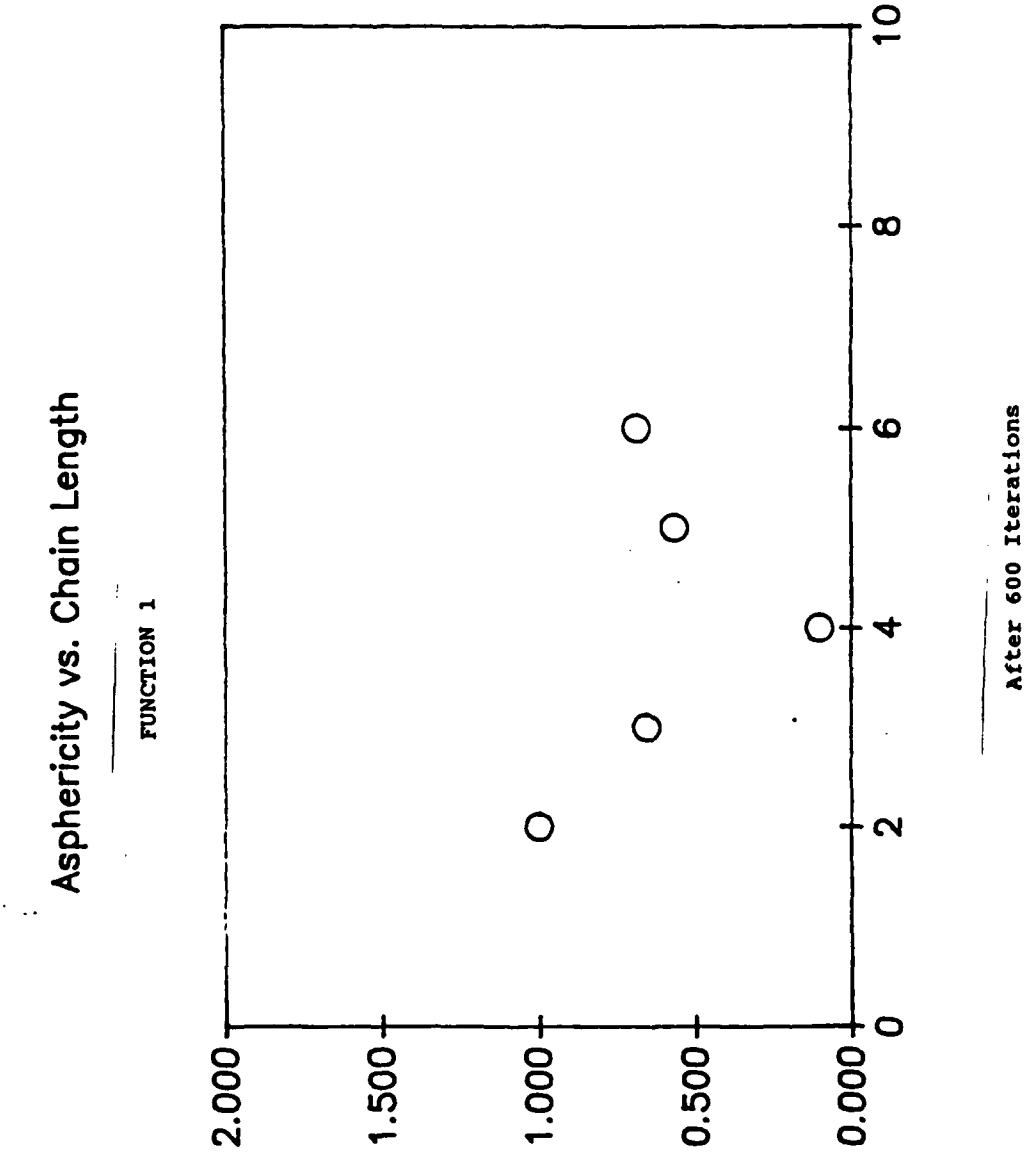

Asphericity vs. Chain Length
FUNCTION 1
2.000
1.500
1.000
0.500
0.000
0
2
4
6
8
10
After 600 Iterations

Figure 25

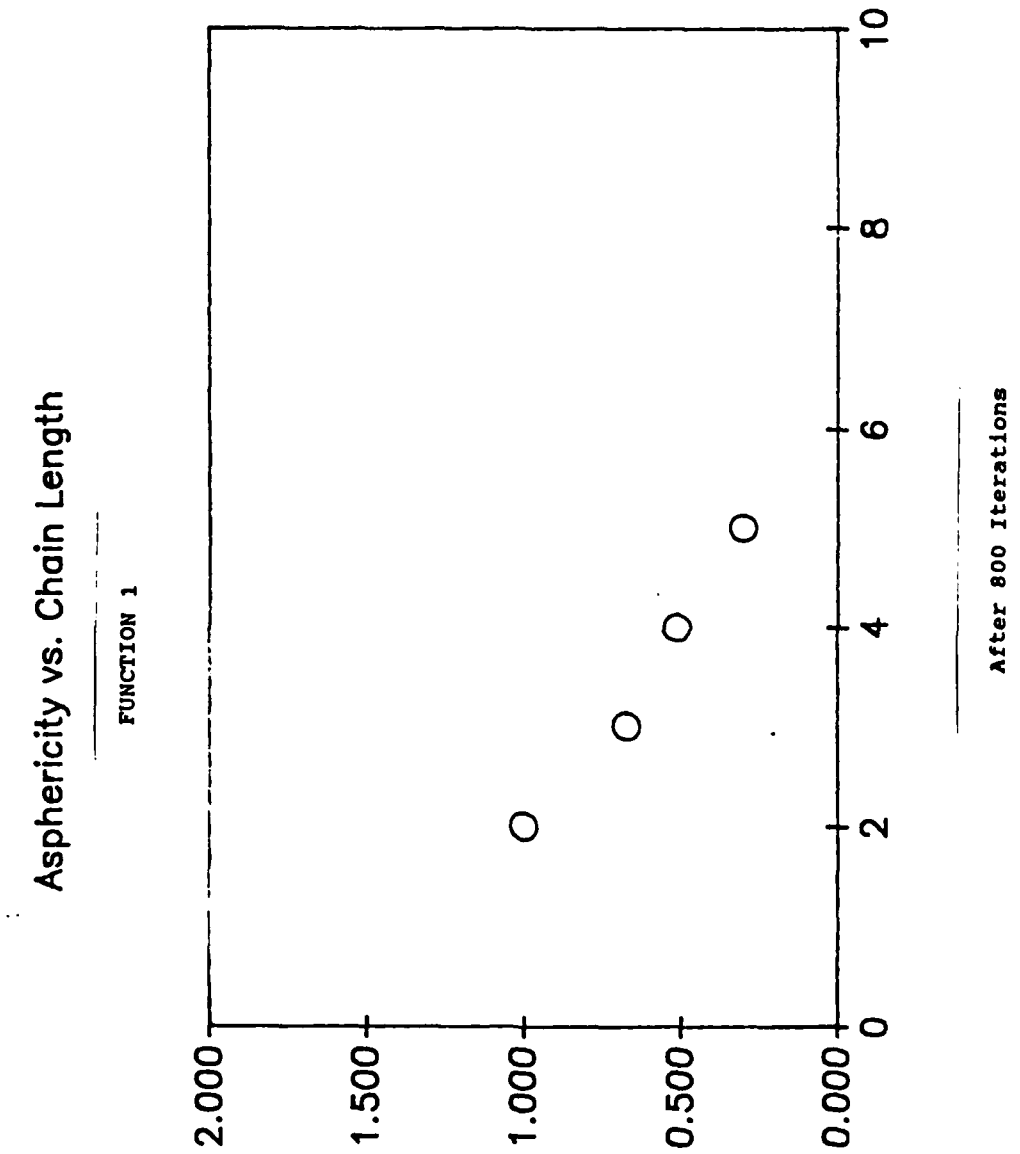

Asphericity vs. Chain Length
FUNCTION 1
2.000
1.500
1.000
0.500
0.000
0
2
4
6
8
10
After 800 Iterations

Figure 26

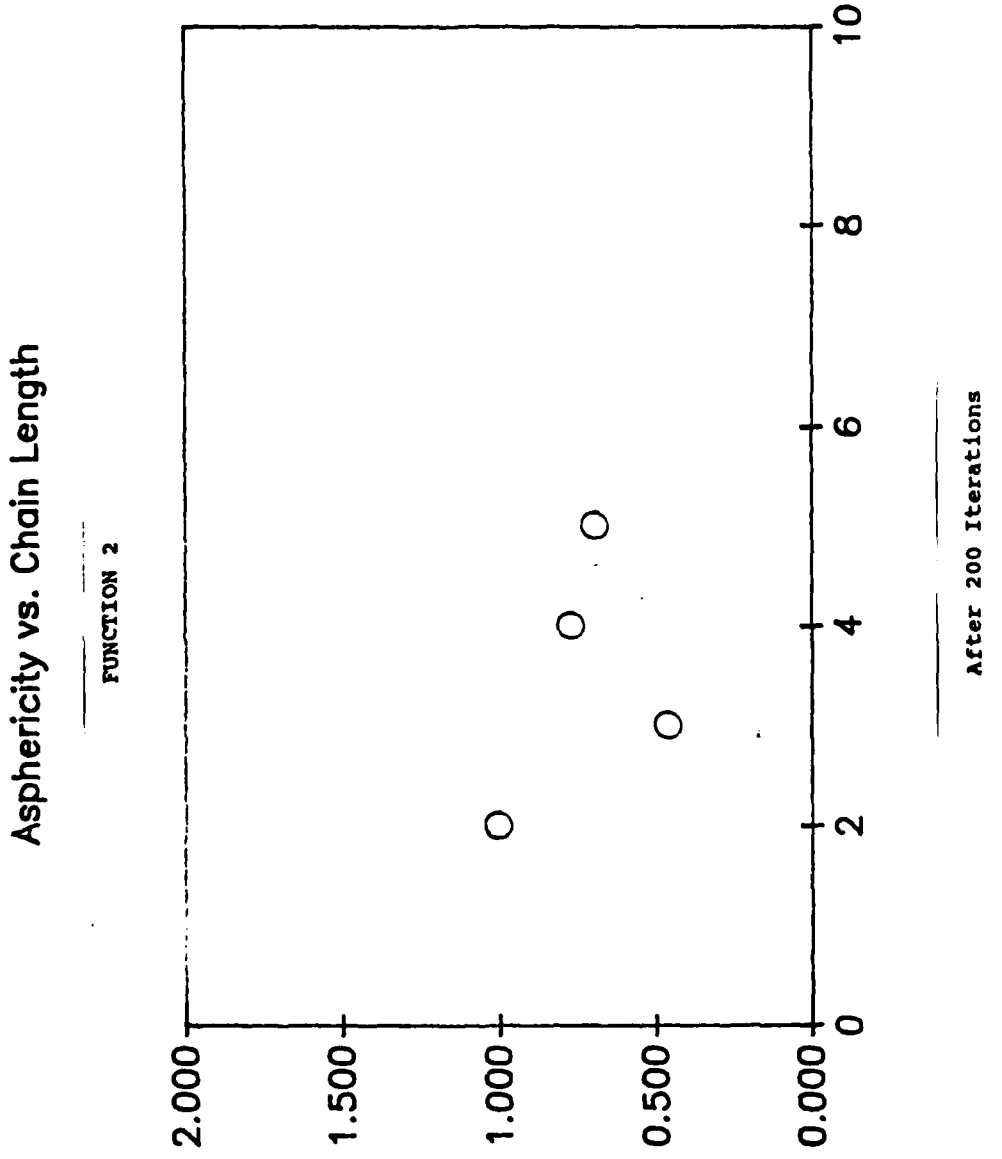

Asphericity vs. Chain Length
FUNCTION 2
After 200 Iterations
2.000
1.500
1.000
0.500
0.000
0
2
4
6
8
10

Figure 27

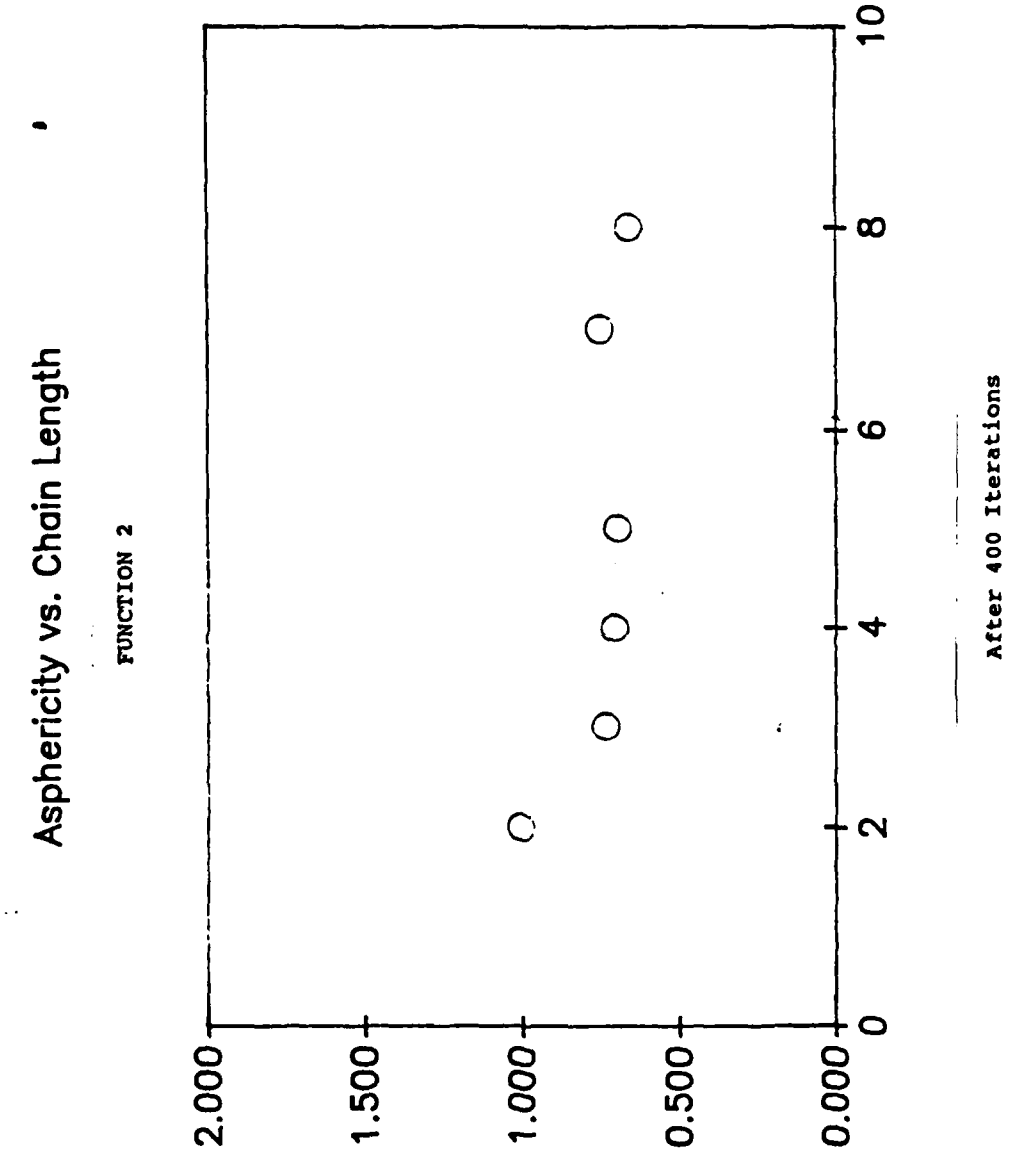

Asphericity vs. Chain Length
FUNCTION 2
2.000
1.500
1.000
0.500
0.000
0
2
4
6
8
10
After 400 Iterations

Figure 28

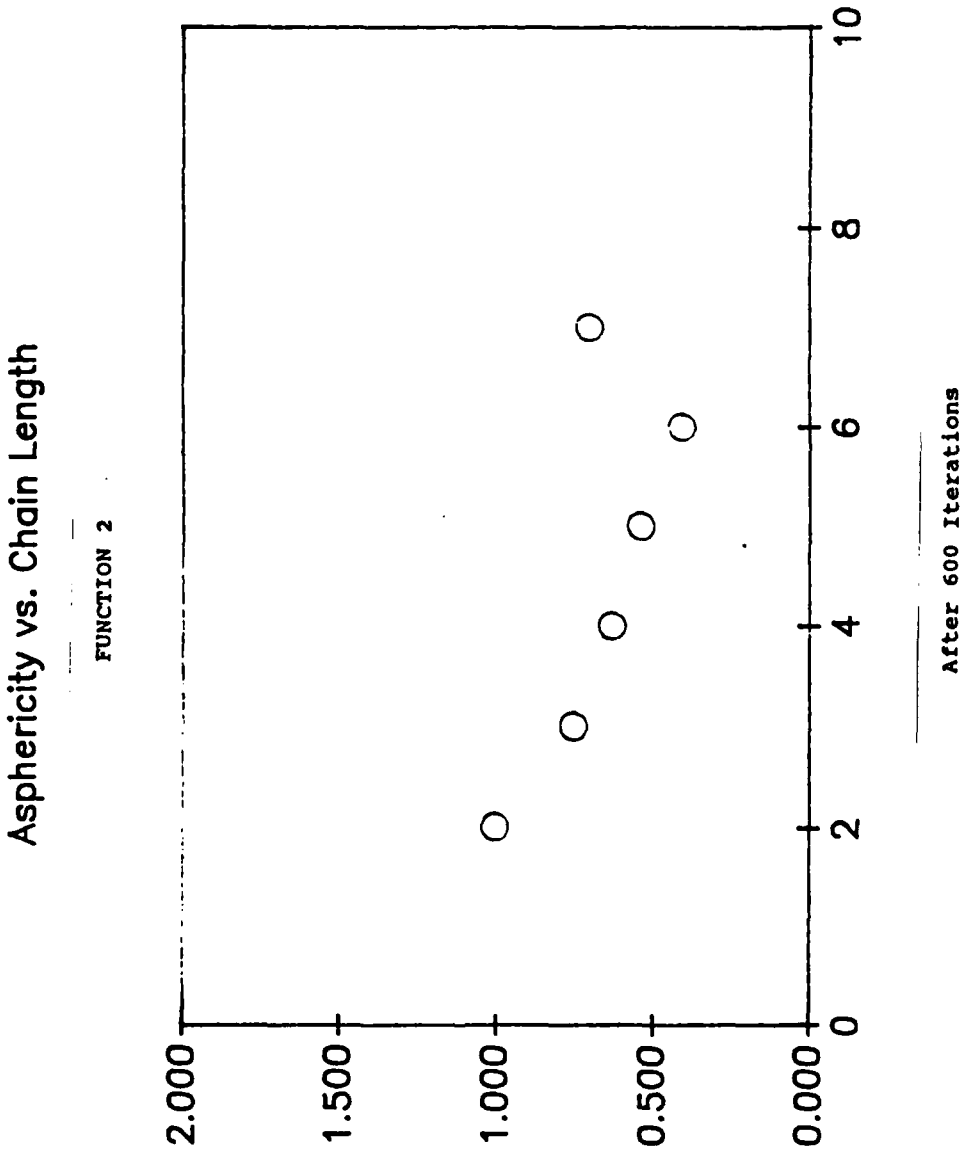
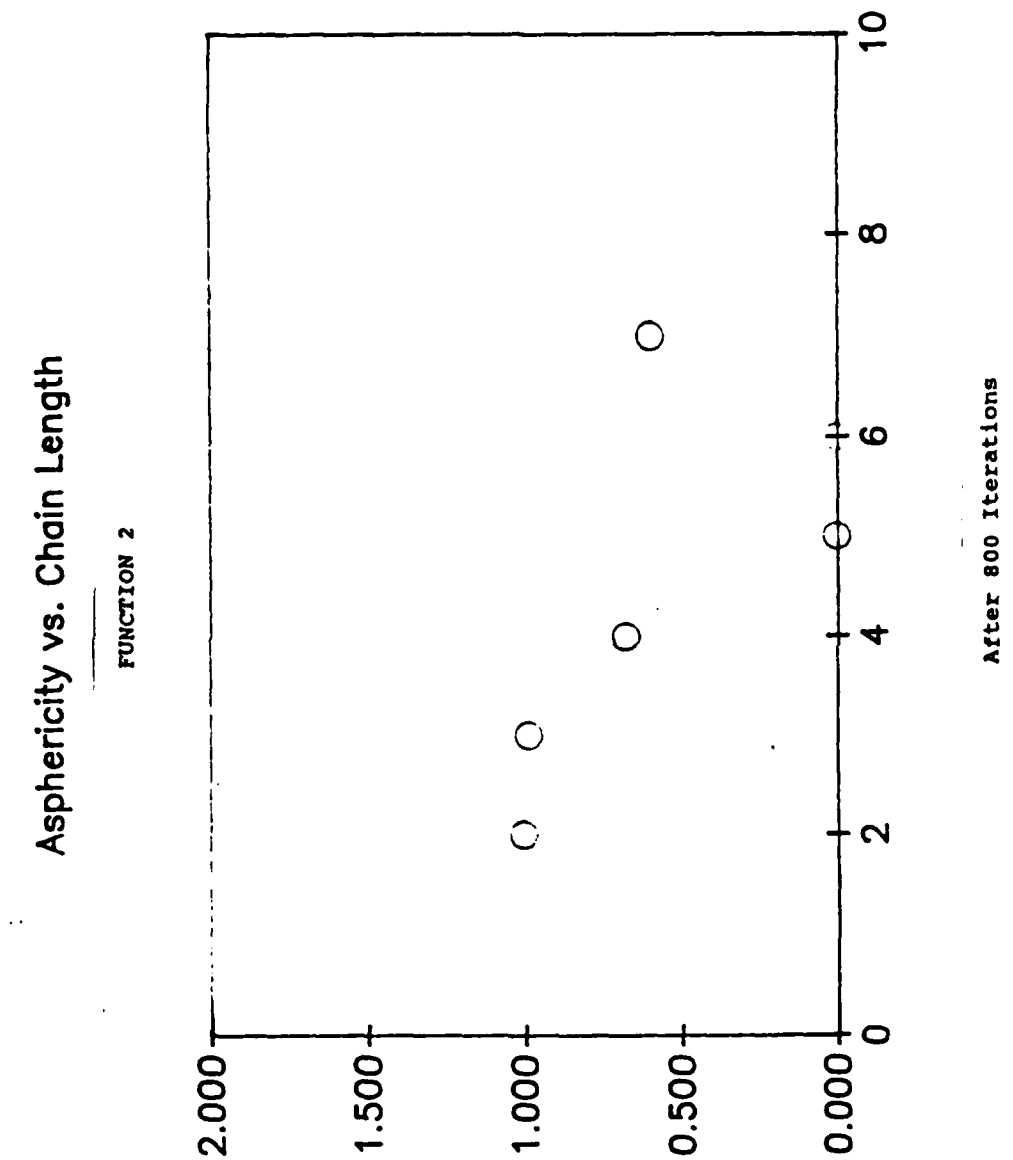

Asphericity vs. Chain Length
FUNCTION 2
2.000
1.500
1.000
0.500
0.000
0
2
4
6
8
10
After 600 Iterations

Figure 29


Asphericity vs. Chain Length
FUNCTION 2
2.000
1.500
1.000
0.500
0.000
0
2
4
6
8
10
After 800 Iterations

receptors. Function (1) resulted in an asphericity of 0.849, while function (2) yielded an asphericity of 0.457 for chains having three receptors. As the probability of the ligand binding to a receptor closer or further than the optimal distance increases, the chains of three become more spherical.

The graphs resulting after 400 iterations (Figures 23, 27) seem to be in a transitional state. In function (1), as the chain length increases by one, the asphericity fluctutates between more linear and more spherical values. The amplitude of this oscillation increases as the chain length increases. During the more linear parts of the oscillation, as the chain length increases the asphericity decreases slightly. This seems intuitive, because as the chains become longer, there is less probability that the receptors will be in a linear configuration. In function (2), the asphericity decreases, levels off, increases slightly, and then decreases again as the chain length increases. The fluctuations in the asphericity in both functions indicate that a transition is occurring from what was a downward trend in asphericity as chain length increased.

After 600 iterations (Figures 24, 28), the asphericity decreases and then increases as the chain length increases for both functions. The difference is the point of transition from increase to decrease in asphericity for

the two functions. For function (1) the minimum asphericity occurs at a chain length of four, while the minimum asphericity for function (2) occurs at a chain length of six. The stronger the ligand binds to receptors closer or further than the optimal distance, the more gradual the changes in aphericity are.

After 800 iterations (Figures 25, 29) the two functions show divergent behavior again. Function (1) gives chains which decrease in asphericity as chain length is increases as a linear function. Function (2), on the otherhand, yields asphericity as a function of chain length that looks like the cosine function from 0 to 135 degrees. The investigation of the relationship between shape and chain length will continue to be an important issue in the study of cross-linking receptors.

SIMULATING THE CONDITIONS OF A B-CELL

The first consideration in modeling the cross-linking between receptors that occurs on the surface of a B-cell is the density of the receptors. To make our model appropriate the following information is considered:

$$\text{Diameter of a B-cell} = 5 \times 10^{-6} \text{ m}$$

$$\text{Radius of a B-cell} = 2.5 \times 10^{-6} \text{ m}$$

$$\text{Surface Area of a B-cell} = 7.85 \times 10^{-11} \text{ m}^2$$

$$\frac{7.85 \times 10^{-11} \text{ m}^2}{100{,}000 \text{ receptors}} = \frac{\text{Area of matrix}}{100 \text{ receptors}}$$

$$\text{Area of matrix} = 7.85 \times 10^{-14} \text{ m}^2$$

$$\text{One side of matrix} = 2.80 \times 10^{-7} \text{ m}$$

$$\text{One side of matrix} = 2800 \text{ Å}$$

For 100 receptors, a matrix that is 2800 Å by 2800 Å is chosen so that the correct density of the receptors on the surface of a B-cell is being modelled. For these runs, cross-linking probability function (2), discussed above, will be employed. Two distinct areas of investigation will be pursued. First, statistical data will be reviewed to validate the results of this application of the Monte Carlo method. Second, trends in chain length and asphericity as a function of the number of iterations performed will be discussed.

Figure 30

TABLE: SIMULATING THE CONDITIONS OF A B-CELL

Chain Length	Number Formed After 200 Iterations									
	Run									
	1	2	3	4	5	6	7	8	9	10
2	7	7	6	7	8	7	7	8	10	9
3	0	1	1	2	1	3	3	2	2	2
4	0	0	0	0	0	0	0	0	0	0

Chain Length	Number Formed After 400 Iterations									
	Run									
	1	2	3	4	5	6	7	8	9	10
2	11	13	12	13	11	10	8	10	11	12
3	2	2	2	1	1	2	2	2	3	2
4	0	0	0	0	1	0	0	0	0	0

Chain Length	Number Formed After 600 Iterations									
	Run									
	1	2	3	4	5	6	7	8	9	10
2	13	15	11	10	9	10	9	8	7	10
3	1	0	2	2	1	1	0	0	1	1
4	0	0	0	0	0	1	1	1	1	1

Chain Length	Number Formed After 800 Iterations									
	Run									
	1	2	3	4	5	6	7	8	9	10
2	10	11	8	9	10	11	14	13	14	13
3	3	3	3	2	3	3	1	1	2	3
4	0	0	1	1	0	0	0	1	1	0

(A) Statistical Data

Ten independent, 800-iteration runs of the program were conducted in which the numbers of each type of chain that resulted after 200, 400, 600, and 800 iterations were recorded (Figure 30). The following table shows the results of those ten runs.

TABLE 1 AFTER 200 ITERATIONS

Chain Length	Average Number	σ_{n-1}	σ_n
1	36.5	4.6	4.3
2	7.6	1.2	1.1
3	1.7	0.9	0.9

TABLE 2 AFTER 400 ITERATIONS

Chain Length	Average Number	σ_{n-1}	σ_n
1	31.2	4.7	4.4
2	11.1	1.5	1.4
3	1.9	0.6	0.5
4	0.1	0.3	0.3

TABLE 3 AFTER 600 ITERATIONS

Chain Length	Average Number	σ_{n-1}	σ_n
1	35.8	3.5	3.3
2	10.2	2.3	2.2
3	0.9	0.7	0.7
4	0.5	0.5	0.5

TABLE 4 AFTER 800 ITERATIONS

Chain Length	Average Number	σ_{n-1}	σ_n
1	31.2	2.9	2.7
2	11.3	2.1	2.0
3	2.4	0.8	0.8
4	0.4	0.5	0.5

The statistical analysis of the data from the ten runs indicates that the Monte Carlo method in this application gives results that are not just particular to a certain sequence of random events. The ten independent runs yielded similar frequencies of chain lengths. Since the standard deviation in most cases was relatively small, the results of a single run are likely to be a close approximation to the average results for many runs.

(B) Chain Length and Asphericity

The chain length as a function of the number of iterations will be discussed first. After 200 iterations, there were 7 chains of two and no chains of any other length (Figure 31). After 400 iterations, 1 chain of three was added (Figure 31). After 800 iterations, another chain of three was added (Figure 31). After 1200 iterations, an additional chain of three was added (Figure 31). After 1600 iterations, an additional chain of two appeared and a chain of three disappeared (Figure 31). After 2400 iterations had passed, 5 more chains of two were added (Figure 31). Finally, after 3200 iterations, three chains of two disappeared for a total of 10 chains of two and 2 chains of three (Figure 31). The system seemed to achieve equilibruim at this point, since the numbers of each type of chain were no longer increasing. The low density of receptor sites used in this run is clearly the reason higher order chains did not form.

The trend in asphericity of chains as a function of the number of iterations is a non-intuitive one that this model can help to elucidate. The asphericity of the chain of two is always one by definition and, therefore, will not be discussed. The average asphericity of the chains of three is 0.974, 0.923, 0.603, 0.436, 0.247, and 0.370 after 400, 600, 800, 1200, 2400, and 3200 iterations,

TABLE: SIMULATING THE CONDITIONS OF A B-CELL

Chain Length	Number Formed						
	Number of Iterations Completed						
	200	400	800	1200	1600	2400	3200
2	7	7	7	7	8	13	10
3	0	1	2	3	2	2	2
4	0	0	0	0	0	0	0

Figure 31

respectively (Figure 32). The average asphericity of the chains of three as a function of the number of iterations, decrease until a minimun is reached and then the asphericity begins to increase again. An analogous trend of decrease, minimum, and increase was found for asphericity as a function of chain length. These two facts tie together if one considers that a greater number of longer chains are formed as the number of iterations increases until an equilibrium is reached.

HIGH DENSITY RECEPTOR PROGRAM RUN

The parameters for this run were purposefully adjusted to double the density of the receptor sites per unit area on the surface of a B-cell. The following values were used:

Number of iterations = 200, 400, 600, 800

Number of receptors = 200

Number of binding sites = 400

Size of Matrix = 900 Å x 900 Å

The frequency of the various sizes of chains that resulted after the specified intervals of iterations are shown in the following table:

TABLE: SIMULATING THE CONDITIONS OF A B-CELL

Chain Length	Asphericity					
	Number of Iterations Completed					
	400	600	800	1200	2400	3200
2	1.00	1.00	1.00	1.00	1.00	1.00
3	.974	.923	.603	.436	.247	.370
4	-	-	-	-	-	-

Figure 32

TABLE:

Chain Length	Number of Iterations			
	200	400	600	800
2	19	25	26	18
3	9	10	8	13
4	4	3	6	9
5	0	5	6	9
6	1	1	1	0
7	0	0	2	0
8	0	0	1	1

The asphericity versus chain length data (Figure 37) after 200, 400, 600, and 800 iterations was graphed to identify any trends that might have occurred. In each of the four graphs (Figures 33-36), as the chain length increased, the asphericity decreased, reached a minimum, and began to increase again. A picture of the receptor cluster chains that had formed after 600 iterations is shown in Figure 38. The receptor cluster loops that formed after 600 iterations are shown in Figure 39. If the trend in asphericity versus chain length is considered in light of the picture of the cross-linking pattern the data make sense. When the chains are small, they tend to be linear

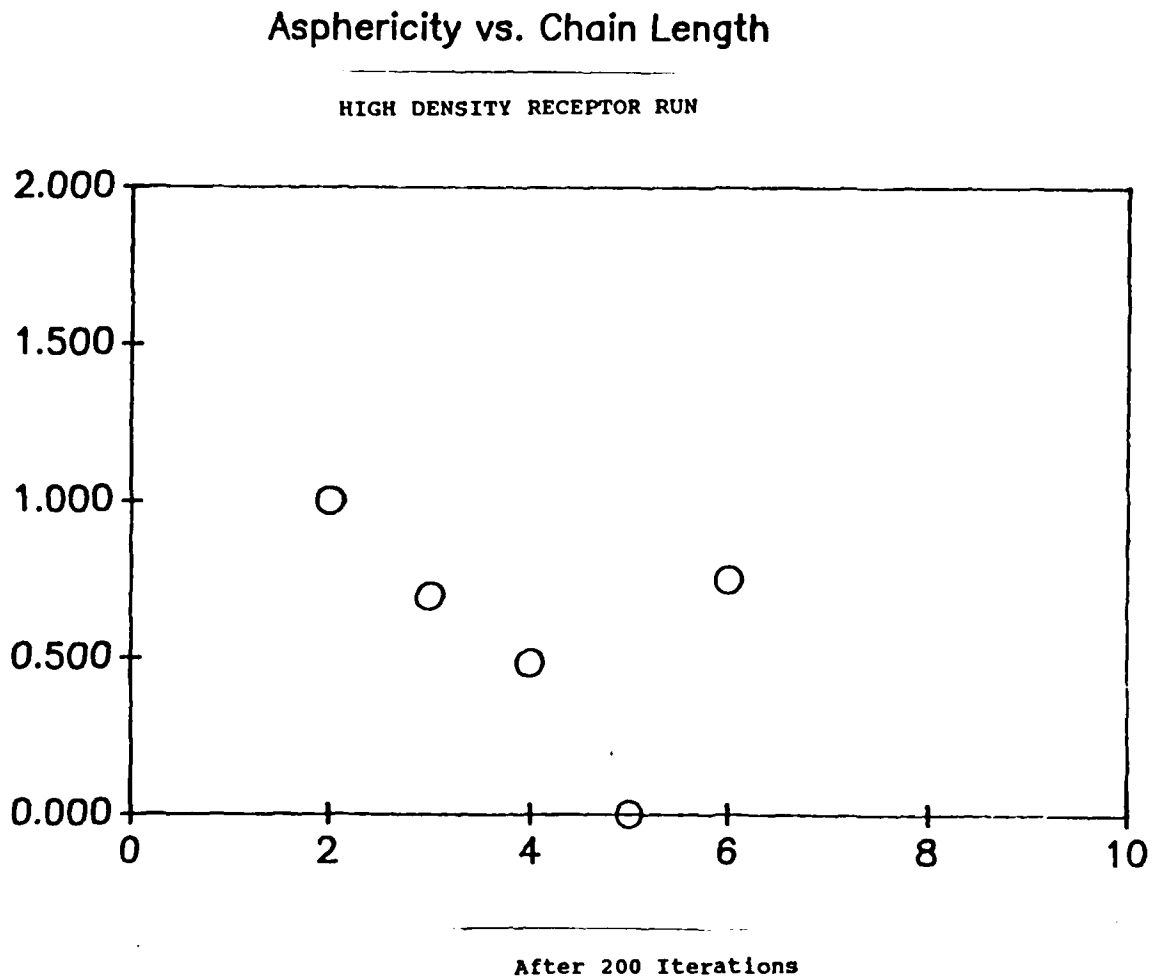

Asphericity vs. Chain Length
HIGH DENSITY RECEPTOR RUN
2.000
1.500
1.000
0.500
0.000
0
2
4
6
8
10
After 200 Iterations

Figure 33

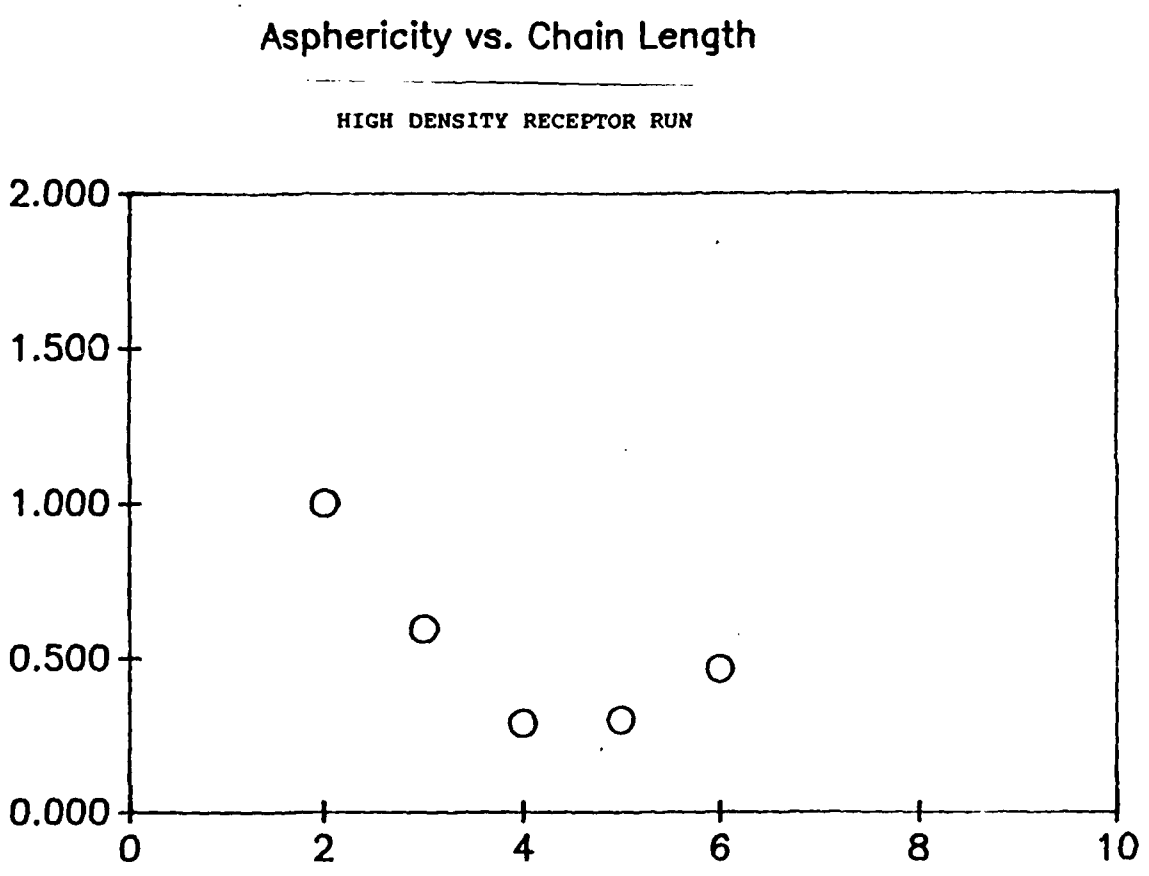
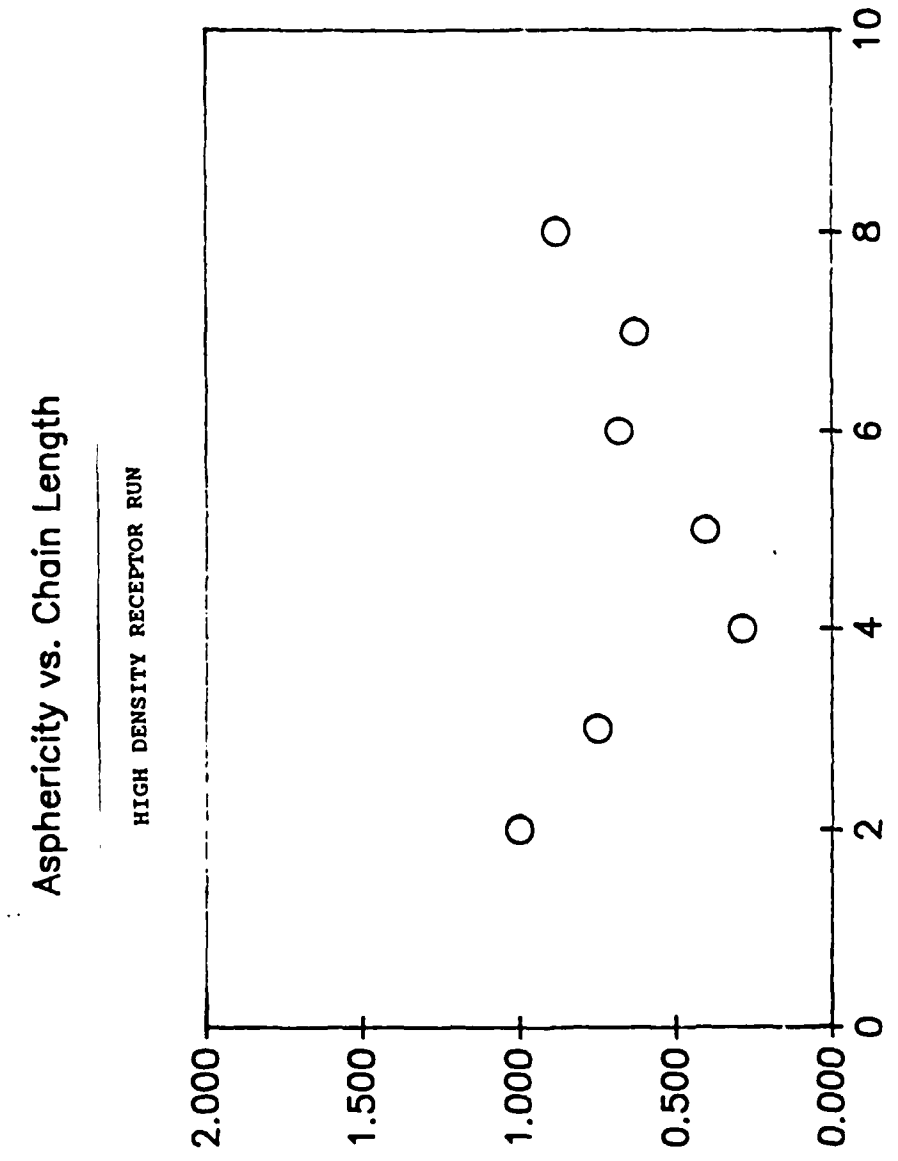

Asphericity vs. Chain Length
HIGH DENSITY RECEPTOR RUN
2.000
1.500
1.000
0.500
0.000
0
2
4
6
8
10
After 400 Iterations

Figure 34

Figure 35


Asphericity vs. Chain Length
HIGH DENSITY RECEPTOR RUN
2.000
1.500
1.000
0.500
0.000
0
2
4
6
8
10

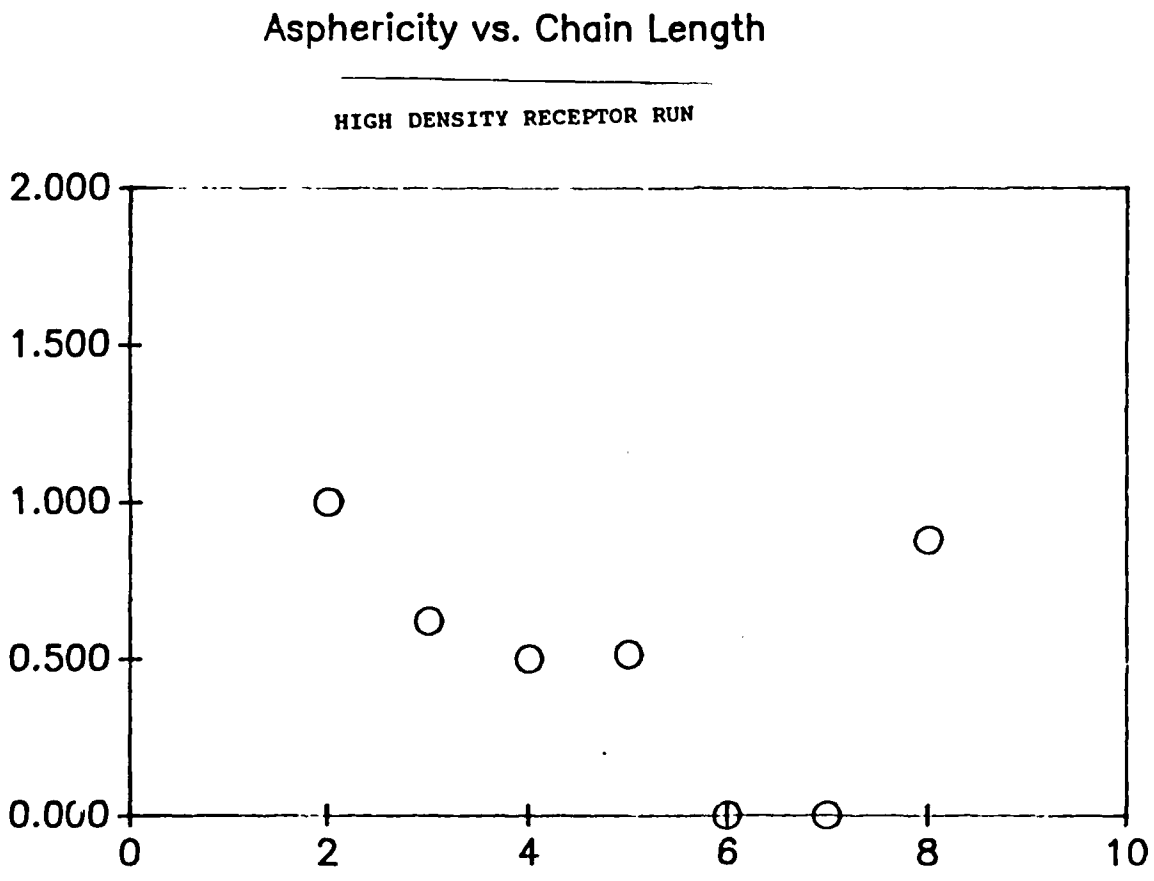

Asphericity vs. Chain Length
HIGH DENSITY RECEPTOR RUN
2.000
1.500
1.000
0.500
0.000
0
2
4
6
8
10
After 800 Iterations

Figure 36

TABLE: HIGH DENSITY RECEPTOR RUN

Chain Length	Asphericity			
	Number of Iterations Completed			
	200	400	600	800
2	1.00	1.00	1.00	1.00
3	.696	.594	.748	.620
4	.483	.287	.286	.501
5	-	.299	.405	.515
6	.750	.463	.680	-
7	-	-	.629	-
8	-	-	.879	.879
9	-	-	-	-

Figure 37

Figure 38

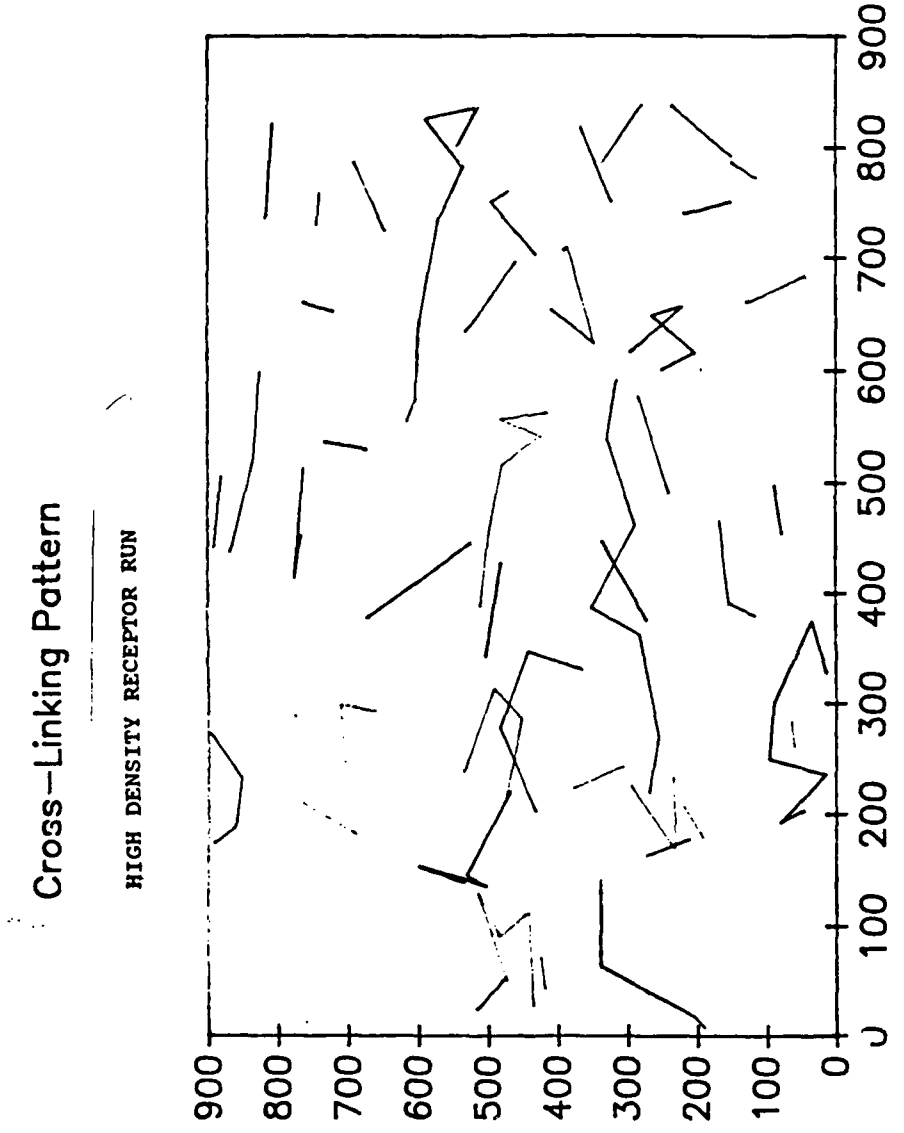
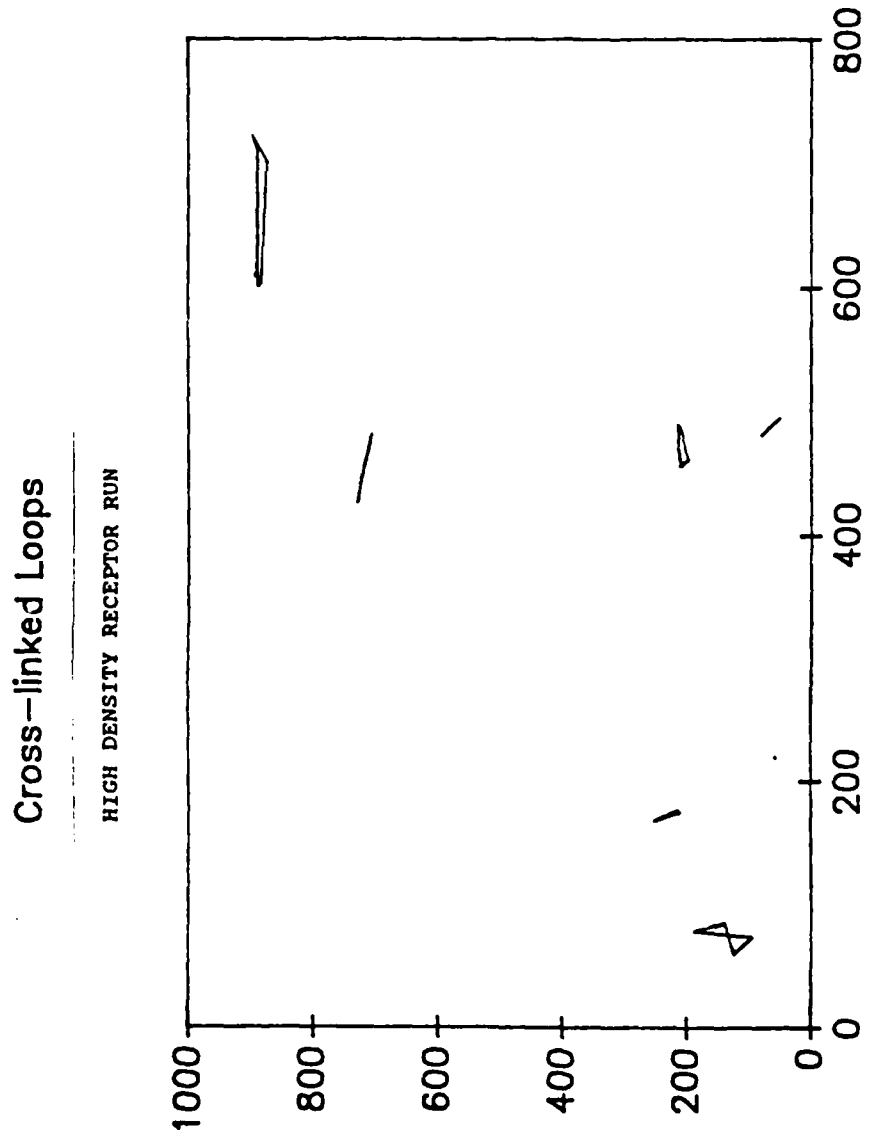

Cross-Linking Pattern
HIGH DENSITY RECEPTOR RUN
0
100
200
300
400
500
600
700
800
900

Figure 39


Cross-linked Loops
HIGH DENSITY RECEPTOR RUN
1000
800
600
400
200
0
0
200
400
600
800

meaning their asphericity is high. When the chains are 4 or 5 receptors long, they tend to be more circular or less aspherical . The chains that are greater than 6 receptors long are more linear in the picture which corresponds to a higher asphericity. Thus the decrease, minimum, and increase in asphericity as the chain length increases from 2 to 8 is intuitive from the picture.

CONCLUSION

While asphericity as a function of cluster size showed certain trends, no evidence was found that would suggest that this shape parameter reaches an equilibrium. Studying the details of the shapes of clusters provides an insight into the mechanism of B-cell activation that goes beyond simple concentration of antigen. It is possible that the shapes of clusters, which continue to be dynamic, control the conformations of enzymes in the B-cell membrane. These enzymes control the release of intracellular calcium and subsequently the activation of the B-cell. This model has shown that the change in asphericity is of the same order of magnitude as the times observed in the oscillations of the intracellular calcium. More research will be required to determine if the shape is a definitive factor in the B-cell activation pathway.

REFERENCES

1. Paul, William E., Fundamental Immunology, Raven Press, New York, 1984, 3.
2. Paul, William E., 4.
3. Paul, William E., 43.
4. Paul, William E., 43-4.
5. Coggeshall, K.M., Monroe, J.G., Ransom, J.T., and Cambier, J.C., Mechanisms of Transmembrane Signal Transduction During B Cell Activation, B-Lymphocyte Differentiation, CRC Press, Inc., Boca Raton, 1986, 3.
6. Waite, B.A. and Chang, E.L., Antibody Multivalency Effects in the Direct Binding Model for Vesicle Immunolysis Assays, J. Immunol. Methods, 115, 231, 1988.
7. Finkelman, F.D., Mond, J.J., and Metcalf, E.S., Antiimmunoglobulin Antibody Induction of B-Lymphocyte Activation in Vivo and in Vitro, B-Lymphocyte Differentiation, CRC Press, Inc., Boca Raton, 1986, 43.
8. Finkelman, F.D., Mond, J.J., and Metcalf, E.S., 42-3.
9. Perelson, A.S., Paradoxes in B-cell Stimulation by

Polymeric Antigen and the Immunon Concept, Paradoxes in Immunology, CRC Press, Inc., Boca Raton, 1986, 200.

10. Perelson, A.S., 208.
11. Perelson, A.S., 209.
12. Wilson, H.A., Greenblatt, D., Poenie, M., Finkelman, F.D., and Tsien, R.Y., Crosslinkage of B lymphocyte Surface Immunoglobulin by Anti-Ig or Antigen Induces Prolonged Oscillation of Intracellular Ionized Calcium, J. Exp. Med., 166, 601, 1987.
13. Rudnick, J. and Gaspari, G., The Shapes of Random Walks, Science, 237, 384, 1987.
14. Porter, R.N. and Raff, L.M., Classical Trajectory Methods in Molecular Collisions, Dynamics of Molecular Collisions, Part B, Plenum Press, New York, 1976, 1.
15. Laidler, K.J., Chemical Kinetics, Harper and Row, New York, 1987, 186.

APPENDIX A

Source Code for Monte Carlo Cross-linking

```
dimension xpt(10000)
dimension ypt(10000)
dimension zpt(10000)
dimension nneigh(10000)
dimension nnval(10000)
dimension ddis(10000)
dimension grph(1000,1000)
dimension neigh(10000)
dimension xpbj(10000)
dimension lue(10000)
dimension nfreq(10000)
dimension low(1000)
dimension high(1000)
dimension nwe(10000)
dimension nnvalue(10000)
dimension ncirc(10000)
dimension nchl(100)
dimension nstatu(100,100)
dimension t(10,10)
dimension tt(10,10)
dimension ttt(10,10)
dimension tradi(1000)
dimension aradi(1000)
dimension asum(1000)
dimension aindex(1000)
dimension aave(1000)
integer npart,c,cc,xcol,ycol,row,b
common/nay/npval
```

```
      common/any/l,ne,nfree,dis(10000),c,nval(10000),cc,
     1  nvalue(10000)
      nlpoh=400
      nglmt=110
      nrlmt=10
      i=1021
      blp=ran(i)
      nfree=200
      n3=0
      nruns=200
11    format(' ',5i5)
      do 14 nerp=1,100
      nchl(nerp)=0
14    continue
      write(50,11) 1
      n0=0
      n1=0
      n2=0
      j=3
      xm=1
      do 300 n=1,nruns
      zz=ran(i)
      xpt(xm)=900*zz
      zz=ran(i)
      ypt(xm)=900*zz
      zz=ran(i)
      kk=2*zz
      zpt(xm)=kk
      xm=xm+1
         if(kk.eq.0)kk=1
         if(kk.eq.1)kk=0
         xpt(xm)=xpt(xm-1)
         ypt(xm)=ypt(xm-1)
         zpt(xm)=kk
```

```
      xm=xm+1
300   continue
      write(50,11) 2
         do 400 n=1,2*nruns
         nvalue(n)=0
400      continue
         write(50,11) 3
      do 6000 nstp=40,43
      nlstp=nstp+70
      ncstp=nstp+60
      ii=1
      do 4150 iii=901,901
      do 1111 aaa=1,nlpoh
      zz=ran(i)
      l=2*nruns*ran(i)+1
      npart=0
      c=1
      dz=nvalue(l)
      d=nvalue(l)
       do 410 n=1,nruns*2
       neigh(n)=0
       nval(n)=0
       dis(n)=0
       nneigh(n)=0
       nnval(n)=0
       ddis(n)=0
410    continue
       write(50,11) 4
         cc=0
         ncc=0
         do 500 n=1,2*nruns
         if(n.eq.l)goto 500
         dist=((xpt(n)-xpt(l))**2 + (ypt(n)-ypt(l))**2)**.5
         if((l+1)/2.eq.l/2)npart=l-1
```

```
      if((l+1)/2.ne.l/2)npart=l+1
      npval=nvalue(npart)
      if(n.eq.npart)goto 500
        if(dist.lt.100)neigh(c)=n
        if(dist.lt.100)nval(c)=nvalue(n)
        if(dist.lt.100)dis(c)=dist
        if(dist.lt.30)nneigh(c)=n
        if(dist.lt.30)nnval(c)=nvalue(n)
        if(dist.lt.30)ddis(c)=dist
      if(dist.gt.30)goto 498
      ncc=ncc+1
498   if(dist.gt.100)goto 500
      cc=c
      c=c+1
500   continue
      write(76,11) (cc,ncc)
511   format(' ',f10.5)
          do 600 c=1,cc
          if(nval(c).eq.0)n0=n0+1
          if(nval(c).eq.1)n1=n1+1
          if(nval(c).eq.2)n2=n2+1
          if(nval(c).gt.2)n3=n3+1
600       continue
          write(50,11) 6
      if(dz.gt.2)d=3
      if(dz.lt.3)goto 620
      if(cc.eq.0)goto 620
      do 610 c=1,cc
      if(nval(c).eq.nvalue(l))ne=c
610   continue
      write(50,11) 7
620   sum=zero(d)+one(d)+twn(d)+twp(d)
      zzz=zero(d)/sum
      ooo=one(d)/sum
```

```
          ttn=twn(d)/sum
          ttp=twp(d)/sum
              zz=ran(i)
              div1=zzz
              div2=zzz+ooo
              div3=zzz+ooo+ttn
          if(zz.gt.div1)goto 700
          jobe=nvalue(1)
          if(jobe.eq.2)nvalue(1)=0
          if(jobe.eq.2)nvalue(npart)=1
          if(jobe.eq.2)goto 1100
          if(jobe.lt.3)goto 660
          if(cc.eq.0)goto 660
              do 650 c=1,cc
              n=neigh(c)
              if(nvalue(1).eq.nval(c))nvalue(n)=1
650           continue
              write(50,11) 8
660           if(nvalue(1).eq.1)nfree=nfree+1
              nvalue(1)=0
              goto 1100
700       if(zz.gt.div2)goto 800
          jobe=nvalue(1)
          if(jobe.eq.2)nvalue(1)=1
          if(jobe.eq.2)nvalue(npart)=0
          if(jobe.eq.2)goto 1100
          if(jobe.lt.3)goto 760
          if(cc.eq.0)goto 760
              do 750 c=1,cc
              n=neigh(c)
              if(nvalue(1).eq.nval(c))nvalue(n)=0
750           continue
              write(50,11) 9
760           if(nvalue(1).eq.0)nfree=nfree-1
```

```
            nvalue(l)=1
            goto 1100
800         if(zz.gt.div3)goto 1000
            smu=0
            if(cc.eq.0)goto 960
                do 950 c=1,cc
                b=nval(c)
                if(b.gt.3)b=3
888             format(' ',e16.8)
                xpbj(c)=doda(b)
                smu=smu+xpbj(c)
950             continue
                write(50,11) 10
960         ww=0
            xx=ran(i)
            bz=0
                do 970 c=1,cc
                if(bz.ne.0)goto 970
                if(smu.eq.0)goto 970
                ww=ww+xpbj(c)/smu
                if(bz.ne.0)goto 970
                if(xx.lt.ww)bz=c
970             continue
                write(50,11) 11
                if(bz.eq.0)goto 1100
                nn=neigh(bz)
                nvalue(nn)=j
                nvalue(l)=j
                if(j.eq.67)write(90,11) (nn,l,cc)
                if(j.eq.67)write(80,11) (nvalue(nml),nml=1,2*nruns)
                j=j+1
                goto 1100
1000        nvalue(l)=2
            nvalue(npart)=2
```

```
1100      rand=ran(i)
          radius=42*rand**3 + -63*rand**2 + 31*rand
          oldx=xpt(1)
          oldy=ypt(1)
            do 1105 c=1,ncc
            angle=asin((xpt(nneigh(c)) - oldx)/((xpt(nneigh(c))
     1      -oldx)**2 + (ypt(nneigh(c))-oldy)**2)**0.5)
          low(c)=angle-.10
          high(c)=angle+.10
1105      continue
          write(50,11) 12
          nxej=0
1106      nflga=0
          angle=ran(i)*2*3.14159
          do 1107 c=1,ncc
          if(angle.gt.low(c).and.angle.lt.high(c))nflga=1
1107      continue
          write(50,11) 13
          if(nflga.eq.1)nxej=nxej+1
          if(nxej.eq.10)goto 1108
          if(nflga.eq.1)goto 1106
1108      xpt(1)=oldx+radius*sin(angle)
          ypt(1)=oldy+radius*cos(angle)
          nflagg=0
          do 1110 c=1,cc
          diss=((xpt(neigh(c))-xpt(1))**2 +
     1          (ypt(neigh(c))-ypt(1))**2)**.5
          if(diss.lt.100)goto 1110
          nflagg=1
1110      continue
          write(50,11) 14
          if(nflagg.ne.1)goto 1111
          xpt(1)=oldx
          ypt(1)=oldy
```

```
 1111     continue
          write(50,11) 15
          do 1200 nli=1,50
          nfreq(nli)=0
 1200     continue
          write(50,11) 16
          n0v=0
          n1v=0
          n2v=0
          q=1
          mch1=0
          do 3000 n=1,2*nruns
             if(nvalue(n).eq.0)n0v=n0v+1
             if(nvalue(n).eq.1)n1v=n1v+1
             if(nvalue(n).eq.2)n2v=n2v+1
             if(nvalue(n).lt.2)lue(q)=n
             if(nvalue(n).lt.2)q=q+1
 3000     continue
          write(50,11) 17
          do 3004 niii=1,10
          do 3003 niiii=1,nglmt
          grph(niii,niiii)=0
 3003     continue
          write(50,11) 18
 3004     continue
          write(50,11) 19
          do 3005 niii=1,2*nruns
          nnvalue(niii)=nvalue(niii)
 3005     continue
          write(50,11) 20
          do 3006 niii=1,nrlmt
          nch1(niii)=0
          ncirc(niii)=0
 3006     continue
```

```
      write(50,11) 21
      xcol=1
      ycol=2
      do 3901 nn=1,2*nruns
      n=nn
      row=1
      nop=0
      if(nnvalue(n).gt.1)goto 3901
3010  if((n+1)/2.eq.n/2)npar=n-1
      if((n+1)/2.ne.n/2)npar=n+1
      if(nnvalue(npar).lt.3)goto 3901
      grph(row,xcol)=xpt(n)
      grph(row,ycol)=ypt(n)
3020                do 3030 nm=1,2*nruns
        if(nnvalue(npar).eq.0)goto 3030
        if(npar.eq.nm)goto 3030
      if(nnvalue(nm).ne.nnvalue(npar))goto 3030
        nnvalue(n)=0
        nnvalue(npar)=0
      n=nm
        if((n+1)/2.eq.n/2)nnpar=n-1
        if((n+1)/2.ne.n/2)nnpar=n+1
      row=row+1
      nop=nop+1
      grph(row,xcol)=xpt(n)
      grph(row,ycol)=ypt(n)
        goto 3040
3030                continue
3040  write(50,11) 22
      nm=1
      npar=nnpar
      if(nnvalue(npar).gt.2)goto 3020
3900        nop=nop+1
            nnvalue(n)=0
```

```
                nnvalue(npar)=0
                nchl(nop)=nchl(nop)+1
              xcol=xcol+2
              ycol=ycol+2
              row=1
              nop=0
3901                    continue
                write(50,11) 23
                nchl(1)=n1v
            xcol=xcol+2
            ycol=ycol+2
            do 4000 nn=1,2*nruns
            n=nn
            row=1
            nop=0
            if(nnvalue(n).lt.3)goto 4000
            if((n+1)/2.eq.n/2)npar=n-1
            if((n+1)/2.ne.n/2)npar=n+1
            if(nnvalue(npar).lt.3)goto 4000
            grph(row,xcol)=xpt(n)
            grph(row,ycol)=ypt(n)
            nstarn=n
3910            do 3990 nm=1,2*nruns
                if(npar.eq.nm)goto 3990
                if(nnvalue(nm).ne.nnvalue(npar))goto 3990
                if(nop.eq.0)goto 3920
                nnvalue(n)=0
                nnvalue(npar)=0
3920            n=nm
                if((n+1)/2.eq.n/2)nnpar=n-1
                if((n+1)/2.ne.n/2)nnpar=n+1
                row=row+1
                nop=nop+1
                grph(row,xcol)=xpt(n)
```

```
                grph(row,ycol)=ypt(n)
                goto 3991
3990            continue
3991          nm=1
              npar=nnpar
              if(n.ne.nstarn)goto 3910
             nnvalue(n)=0
             nnvalue(npar)=0
             nchl(nop)=nchl(nop)+1
             xcol=xcol+2
             ycol=ycol+2
4000         continue
             do 4005 n=1,1000
             tradi(n)=0
             aradi(n)=0
             asum(n)=0
             aindex(n)=0
4005         continue
             xcol=1
             ycol=2
4010         tt(1,1)=0
             tt(1,2)=0
             tt(2,1)=0
             tt(2,2)=0
             xm=0
             ym=0
             row=1
4015         xm=xm+grph(row,xcol)
             ym=ym+grph(row,ycol)
             row=row+1
             if(grph(row,xcol).ne.0)goto 4015
             if(grph(row,ycol).ne.0)goto 4015
             row=row-1
             xm=xm/row
```

```
      ym=ym/row
      row=1
4020  t(1,1)=(grph(row,xcol)-xm)*(grph(row,xcol)-xm)
      t(1,2)=(grph(row,xcol)-xm)*(grph(row,ycol)-ym)
      t(2,1)=(grph(row,ycol)-ym)*(grph(row,xcol)-xm)
      t(2,2)=(grph(row,ycol)-ym)*(grph(row,ycol)-ym)
      tt(1,1)=tt(1,1)+t(1,1)
      tt(1,2)=tt(1,2)+t(1,2)
      tt(2,1)=tt(2,1)+t(2,1)
      tt(2,2)=tt(2,2)+t(2,2)
      row=row+1
      if(grph(row,xcol).ne.0)goto 4020
      if(grph(row,ycol).ne.0)goto 4020
      row=row-1
      ttt(1,1)=tt(1,1)/row
      ttt(1,2)=tt(1,2)/row
      ttt(2,1)=tt(2,1)/row
      ttt(2,2)=tt(2,2)/row
      tradi(xcol)=((ttt(1,1)+ttt(2,2)+(((ttt(1,1)+ttt(2,2))**2)
     1  -(4*(ttt(1,1)*ttt(2,2)-ttt(2,1)*ttt(1,2))))**0.5)/2)
      if(tradi(xcol).lt.0)tradi(xcol)=0
      tradi(xcol)=(tradi(xcol))**0.5
      tradi(ycol)=((ttt(1,1)+ttt(2,2)-(((ttt(1,1)+ttt(2,2))**2)
     1  -(4*(ttt(1,1)*ttt(2,2)-ttt(2,1)*ttt(1,2))))**0.5)/2)
      if(tradi(ycol).lt.0)tradi(ycol)=0
      tradi(ycol)=(tradi(ycol))**0.5
      aradi(xcol)=((tradi(xcol)**2 - tradi(ycol)**2)**2)/
     1  (tradi(xcol)**2 + tradi(ycol)**2)**2
      asum(row)=asum(row)+aradi(xcol)
      aindex(row)=aindex(row)+1
      write(ncstp,4026) (aradi(xcol),row)
      write(nlstp,4025) (tradi(xcol),tradi(ycol))
      xcol=xcol+2
      ycol=ycol+2
```

```
      if(grph(2,xcol).ne.0)goto 4010
      if(grph(2,ycol).ne.0)goto 4010
      nastp=nstp+11
      do 4024 n=1,10
      if(aindex(n).eq.0)aave(n)=0
      if(aindex(n).eq.0)goto 4023
      aave(n)=asum(n)/aindex(n)
4023  write(nastp,4027) (asum(n),aindex(n),aave(n))
4024  continue
4025  format(' ',2f10.5)
4026  format(' ',1f10.5,i8)
4027  format(' ',3f10.5)
      do 4108 n=1,nrlmt
      nstatu(n,ii)=nchl(n)
4108  continue
      write(50,11) 24
4109  format(' ',10I3)
4111  format(' ',5I5)
4112  format(' ',2f10.5,i8)
      nnstp=nstp-40
      do 4120 n=1,nrlmt
      write(nnstp,4113) (grph(n,m),m=1,nglmt)
4113  format(' ',10f10.5)
4120  continue
      write(50,11) 25
4130  ii=ii+1
4150  continue
      write(50,11) 26
      do 4160 n=1,10
      write(nstp,4109) (nstatu(n,m),m=1,10)
4160  continue
      write(50,11) 27
6000  continue
      write(50,11) 28
```

```
      stop
      end

      function doda(b)
      integer c,cc,b
      common/any/1,ne,nfree,dis(10000),c,nval(10000),cc,
     1   nvalue(10000)
      nor=0
      mor=0
      if(nvalue(1).eq.1)mor=0
      if(nvalue(1).eq.1)nor=1
      if(nvalue(1).eq.0)nor=0
      if(nvalue(1).eq.0)mor=1
34    format(' ',e16.8)
      if(b.eq.0)doda=nor*.8*exp(-.0003*(dis(c)-50)**2)
      if(b.eq.1)doda=mor*.8*exp(-.0003*(dis(c)-50)**2)
      if(b.eq.2)doda=0
      if(b.eq.3)doda=0
      return
      end
         function zero(d)
         integer c,cc
         common/any/1,ne,nfree,dis(10000),c,nval(10000)
     1     ,cc,nvalue(10000)
         if(d.eq.0)zero=.10
         if(d.eq.1)zero=.30
         if(d.eq.2)zero=.30
         if(d.eq.3)zero=.9-.8*exp(-.0003*(dis(ne)-50)**2)
         return
         end
      function one(d)
      integer c,cc
      common/any/1,ne,nfree,dis(10000),c,nval(10000)
     1  ,cc,nvalue(10000)
```

```
      if(d.eq.0)one=.005*nfree
      if(d.eq.1)one=.10

      if(d.eq.2)one=.30
      if(d.eq.3)one=.9-.8*exp(-.0003*(dis(ne)-50)**2)
      return
      end
         function twn(d)
         integer c,cc
         common/any/l,ne,nfree,dis(10000),c,nval(10000)
     1    ,cc,nvalue(10000)
         if(d.eq.0)m=1
         if(d.eq.0)twn=smuu(m)
         if(d.eq.1)m=0
         if(d.eq.1)twn=smuu(m)
         if(d.eq.2)twn=0
         if(d.eq.3)twn=.9
         return
         end
      function twp(d)
      integer c,cc
      common/nay/npval
      common/any/l,ne,nfree,dis(10000),c,nval(10000)
     1  ,cc,nvalue(10000)
      nor=0
      mor=0
      if(npval.eq.1)nor=1
      if(npval.eq.0)nor=0
      if(npval.eq.1)mor=0
      if(npval.eq.0)mor=1
      if(d.eq.0)twp=nor
      if(d.eq.1)twp=mor
      if(d.eq.2)twp=.60
      if(d.eq.3)twp=0
```

```
      return
      end
         function smuu(m)
         integer c,cc
         common/any/l,ne,nfree,dis(10000),c,nval(10000)
     1    ,cc,nvalue(10000)
         smuu=0
         do 5000 c=1,cc
         if(nval(c).ne.m)goto 5000
         smuu=smuu+.8*exp(-.0003*(dis(c)-50)**2)
5000     continue
         return
         end
```